arXiv:1210.0828v1 [cs.LO] 2 Oct 2012

# Data types with symmetries and polynomial functors over groupoids

Joachim Kock[1,2]

*Departament de matemàtiques*
*Universitat Autònoma de Barcelona*
*Spain*

**Abstract**

Polynomial functors (over **Set** or other locally cartesian closed categories) are useful in the theory of data types, where they are often called containers. They are also useful in algebra, combinatorics, topology, and higher category theory, and in this broader perspective the polynomial aspect is often prominent and justifies the terminology. For example, Tambara's theorem states that the category of finite polynomial functors is the Lawvere theory for commutative semirings [45], [18].

In this talk I will explain how an upgrade of the theory from sets to groupoids (or other locally cartesian closed 2-categories) is useful to deal with data types with symmetries, and provides a common generalisation of and a clean unifying framework for quotient containers (in the sense of Abbott et al.), species and analytic functors (Joyal 1985), as well as the stuff types of Baez and Dolan. The multi-variate setting also includes relations and spans, multispans, and stuff operators. An attractive feature of this theory is that with the correct homotopical approach — homotopy slices, homotopy pullbacks, homotopy colimits, etc. — the groupoid case looks exactly like the set case.

After some standard examples, I will illustrate the notion of data-types-with-symmetries with examples from perturbative quantum field theory, where the symmetries of complicated tree structures of graphs play a crucial role, and can be handled elegantly using polynomial functors over groupoids. (These examples, although beyond species, are purely combinatorial and can be appreciated without background in quantum field theory.)

Locally cartesian closed 2-categories provide semantics for a 2-truncated version of Martin-Löf intensional type theory. For a fullfledged type theory, locally cartesian closed $\infty$-categories seem to be needed. The theory of these is being developed by David Gepner and the author as a setting for homotopical species, and several of the results exposed in this talk are just truncations of $\infty$-results obtained in joint work with Gepner. Details will appear elsewhere.

*Keywords:* Polynomial functors, groupoids, data types, symmetries, species, trees.

[1]Email: kock@mat.uab.cat

[2]The author has profited greatly from related collaborations with André Joyal, Nicola Gambino, Anders Kock, Imma Gálvez, Andy Tonks, and notably David Gepner. Partial support from research grants MTM2009-10359 and MTM2010-20692 of Spain is gratefully acknowledged.

# 1  Polynomial functors over *Set* and data types

**1.1 Polynomial functors in one variable.** In its simplest form, a *polynomial functor* is an endofunctor of *Set* of the form

$$X \mapsto \sum_{b \in B} X^{E_b}. \tag{1}$$

Here the sum sign is disjoint union of sets, $X^{E_b}$ denotes the hom set $\mathrm{Hom}(E_b, X)$, and $(E_b \mid b \in B)$ is a $B$-indexed family of sets, encoded conveniently as a single map of sets

$$E \to B.$$

Viewed as a data type constructor, $E \to B$ is often called a *container* [1, 2, 3, 4, 5, 7]; then $B$ is regarded as a set of shapes, and the fibre $E_b$ is the set of positions in the shape corresponding to $b$. The data to be inserted into these positions can be of any type $X$: the polynomial functor receives a type $X$ (a set) and returns the new more elaborate type $\sum X^{E_b}$. Polymorphic functions correspond to natural transformations of polynomial functors, and these can be handled in terms of the representing sets $E \to B$ alone, cf. [1], [18], and 2.6 below. A fundamental example is the list functor, $X \mapsto \sum_{n \in \mathbb{N}} X^{\mathbf{n}}$, which to a set $X$ associates the set of lists of elements in $X$. Here $n \in \mathbb{N}$ is the shape, and $\mathbf{n}$ denotes the $n$-element set $\{0, 1, \ldots, n-1\}$ of positions in a length-$n$ list.

There is another important use of polynomial functors in type theory: one then regards $E \to B$ as a signature generating an algebra, namely the initial algebra for the polynomial functor. Initial algebras for polynomial functors are inductive data types, corresponding to W-types in (extensional) Martin-Löf type theory [42], [40]. Similarly, terminal coalgebras are coinductive data types (sometimes called M-types), often interpreted as programs or systems (see for example [43], [23]).

**1.2 Species and analytic functors.** A functor is *finitary* when it preserves $\omega$-filtered colimits. For a polynomial functor this is equivalent to $E \to B$ having finite fibres. Let $\mathbb{B}_\omega$ denote the groupoid of finite sets and bijections. A *species* [26] is a functor $F : \mathbb{B}_\omega \to \textbf{\textit{Set}}$, written $S \mapsto F[S]$; the set $F[S]$ is to be thought of as the set of $F$-structures that can be put on the set $S$. The *extension* of $F$ is the endofunctor

$$
\begin{aligned}
\textbf{\textit{Set}} &\longrightarrow \textbf{\textit{Set}} \\
X &\longmapsto \sum_{n \in \mathbb{N}} \frac{F[n] \times X^n}{\mathrm{Aut}(n)}
\end{aligned}
\tag{2}
$$

which is the left Kan extension of $F$ along the (non-full) inclusion $\mathbb{B}_\omega \subset \mathbf{Set}$. A functor of this form is called *analytic* [27]. Joyal established an equivalence of categories between species and analytic functors, and characterised analytic functors as the finitary functors preserving cofiltered limits and weak pullbacks [27], see also [24] and [6]. Finitary polynomial functors are precisely the analytic functors which preserve pullbacks strictly. In terms of species they correspond to those for which the symmetric group actions are free.

Monoids in species (under the operation of substitution, which corresponds to composition of analytic functors) are precisely operads. Many important polynomial functors have the structure of monad. For example, the list functor has a natural monad structure by concatenation of lists. Polynomial monads equipped with a cartesian monad map to the list monad are the same thing as non-symmetric operads [37]. More generally, finitary polynomial monads correspond to projective operads [32] (i.e. such that every epi to it splits).

**1.3 Polynomial functors in many variables.** Following [18], a *polynomial* is a diagram of sets

$$I \xleftarrow{s} E \xrightarrow{p} B \xrightarrow{t} J, \tag{3}$$

and the associated *polynomial functor* (or the *extension* of the polynomial) is given by the composite

$$\mathbf{Set}_{/I} \xrightarrow{\Delta_s} \mathbf{Set}_{/E} \xrightarrow{\Pi_p} \mathbf{Set}_{/B} \xrightarrow{\Sigma_t} \mathbf{Set}_{/J}, \tag{4}$$

where $\Delta_s$ is pullback along $s$, $\Pi_p$ is the right adjoint to pullback (called dependent product), and $\Sigma_t$ is left adjoint to pullback (called dependent sum). For a map $f : B \to A$ we have the three explicit formulae

$$\Delta_f(X_a \mid a \in A) = (X_{f(b)} \mid b \in B) \tag{5}$$

$$\Sigma_f(Y_b \mid b \in B) = (\sum_{b \in B_a} Y_b \mid a \in A) \tag{6}$$

$$\Pi_f(Y_b \mid b \in B) = (\prod_{b \in B_a} Y_b \mid a \in A), \tag{7}$$

giving altogether the following formula for (4)

$$(X_i \mid i \in I) \longmapsto (\sum_{b \in B_j} \prod_{e \in E_b} X_{s(e)} \mid j \in J),$$

which specialises to (1) when $I = J = 1$.

The multi-variate polynomial functors correspond to *indexed containers* [7], and their initial algebras are sometimes called *general tree types* [41, Ch. 16].

From the abstract description in terms of adjoints, it follows that the notion of polynomial functor (and most of the theory) makes sense in any locally cartesian closed category, and polynomial functors are the most natural class

of functors between slices of such categories. They have been characterised intrinsically [31] as the local fibred right adjoints.

**1.4 Incorporating symmetries.** A container is a rigid data structure: it does not allow for data to be permuted in any way among the positions of a given shape. In many situations it is desirable to allow for permutation, so that certain positions within a shape become indistinguishable. In quantum physics, the principle of indistinguishable particles imposes such symmetry at a fundamental level. A fundamental example is the multiset data type, whose extension is the functor

$$X \mapsto \sum_{n \in \mathbb{N}} \frac{X^{\mathbf{n}}}{\mathrm{Aut}(\mathbf{n})}, \tag{8}$$

which is analytic but not polynomial.

In order to account for such data types with symmetries, Abbott et al. [5] (see also Gylterud [22]) have extended the container formalism by adding the symmetries 'by hand': for each shape (element $b$ in $B$) there is now associated a group of symmetries of the fibre $E_b$, and data inserted into the corresponding positions is quotiented out by this group action. It is not difficult to see (cf. also [6]) that in the finitary case, this is precisely the notion of species and analytic functors.

In fact it has been in the air for some time (see for example [14], and more recently [12], [47]) that species should be a good framework for data type theory. It is the contention of the present contribution that polynomial functors over groupoids provide a clean unifying framework: in the setting of groupoids, the essential distinction between 'analytic' and 'polynomial' evaporates (3.7), and the functors can be represented by diagrams with combinatorial content (3) just as polynomials over sets, as we proceed to explain.

From the viewpoint of species, there are other reasons for this upgrade anyway. In fact, it was soon realised by combinatorists that the 1985 notion of analytic functors is not optimal for enumerative purposes: taking cardinality simply does not yield the exponential generating functions central to enumerative combinatorics! (It does so if the analytic functor is polynomial.) In fact, the Species Book [11] does not mention analytic functors at all.

The issue was sorted out by Baez and Dolan [9]: the problem is that dividing out by the group action in (2) is a bad quotient from the viewpoint of homotopy theory, and does not behave well with respect to cardinality. To get the correct cardinalities, it is necessary to use *homotopy quotients*, and the result is then no longer a set but a groupoid, and the cardinality has to be *homotopy cardinality*. So it is necessary to work from the beginning with groupoids instead of sets. Baez and Dolan introduced *species in groupoids* (3.6), dubbing them *stuff types*, showed that homotopy cardinality gives the correct generating functions, and illustrated the usefulness of the broader generality by showing how the types needed for a combinatorial description of the

quantum harmonic oscillator are stuff types, not classical species [9].

Joint work with David Gepner closes the circle by observing that over groupoids, species/analytic functors are the same thing as discrete finitary polynomial functors (3.7); hence the neat formalism of polynomials provides a natural unifying framework for (quotient) containers and species.

# 2 Polynomial functors over groupoids

A groupoid is a category in which all arrows are invertible. A useful intuition for the present purposes is that groupoids are 'sets fattened with symmetries'. From the correct homotopical viewpoint groupoids behave very much like sets. We are interested in groupoids up to equivalence, and for this reason many familiar 1-categorical notions, such as pullback and fibre, are not appropriate, as they are not invariant under equivalence. The good notions are the corresponding *homotopy* notions, which we briefly recall. They can all be deduced from the beautiful simplicial machinery developed by Joyal [28, 29] to generalise the theory of categories to quasi-categories (called $\infty$-categories by Lurie [38]). Since the 2-category **Grpd** of groupoids has only invertible 2-cells, it is an example of a quasi-category. From now on when we say 2-category we shall mean '2-category with only invertible 2-cells'.

**2.1 Slices.** If $I$ is a groupoid, the *homotopy slice* **Grpd**$_{/I}$ is the 2-category of *projective cones* with base $I$ (cf. [28]): its objects are maps $X \to I$; its arrows are triangles with a 2-cell

$$
\begin{array}{ccc}
X & \longrightarrow & Y \\
& \searrow \;\Rightarrow\; \swarrow & \\
& I &
\end{array}
$$

and 2-arrows are diagrams

$$
\begin{array}{ccc}
X & \underset{\Downarrow}{\rightrightarrows} & Y \\
& \searrow \;\Rightarrow\Rightarrow\; \swarrow & \\
& I &
\end{array}
$$

commuting with the structure triangles. More generally, if $d : T \to$ **Grpd** is any diagram, there is a 2-category **Grpd**$_{/d}$ of projective cones with base $d$.

A *homotopy terminal object* in a 2-category $\mathscr{C}$ is an object $t$ such that for any other object $x$, the groupoid $\mathscr{C}(x, t)$ is contractible, i.e. equivalent to a point. More general homotopy limits are defined in the usual way using homotopy slices: the homotopy limit of a functor $d : T \to$ **Grpd** is by definition a homotopy terminal object in the homotopy slice **Grpd**$_{/d}$. Homotopy limits are unique up to equivalence.

**2.2 Pullbacks and fibres.** Given a diagram of groupoids $X, Y, S$ indicated by the solid arrows,

$$
\begin{array}{ccc}
X \times_S Y & \dashrightarrow & Y \\
\downarrow & \lrcorner & \downarrow{\scriptstyle g} \\
X & \xrightarrow{\ f\ } & S
\end{array}
$$

the *homotopy pullback* is the homotopy limit, i.e. given as a homotopy terminal object in a a certain slice 2-category of projective cones over the solid diagrams of the shape in question, and as such it is determined uniquely up to equivalence. A specific model is the groupoid $X \times_S Y$ whose objects are triples $(x, y, \phi)$ with $x \in X$, $y \in Y$ and $\phi : fx \to gy$ an arrow of $S$, and whose arrows are pairs $(\alpha, \beta) : (x, y, \phi) \to (x', y', \phi')$ consisting of $\alpha : x \to x'$ an arrow in $X$ and $\beta : y \to y'$ an arrow in $Y$ such that the following diagram commutes in $S$

$$
\begin{array}{ccc}
fx & \xrightarrow{\ \phi\ } & gy \\
{\scriptstyle f(\alpha)}\downarrow & & \downarrow{\scriptstyle g(\beta)} \\
fx' & \xrightarrow{\ \phi'\ } & gy'.
\end{array}
$$

(One should note that if $f$ or $g$ is a fibration then the naïve set-theoretic pullback is equivalent to the homotopy pullback.)

The *homotopy fibre* $E_b$ of a morphism $p : E \to B$ over an object $b$ in $B$ is the homotopy pullback of $p$ along the inclusion map $1 \xrightarrow{\ \ulcorner b \urcorner\ } B$ :

$$
\begin{array}{ccc}
E_b & \longrightarrow & E \\
\downarrow & \lrcorner & \downarrow{\scriptstyle p} \\
1 & \xrightarrow{\ \ulcorner b \urcorner\ } & B.
\end{array}
$$

(Note that the homotopy fibre $E_b$ is not in general a subgroupoid of $E$, although the map $E_b \to E$ is always faithful. But again, if $p$ is a fibration then the set-theoretic fibre is equivalent to the homotopy fibre.)

**2.3 Homotopy quotients.** Whenever a group $G$ acts on a set or a groupoid $X$, the *homotopy quotient* $X/G$ is the groupoid obtained by gluing in a path (i.e. an arrow) between $x$ and $y$ for each $g \in G$ such that $gx = y$. More formally it is the total space of the Grothendieck construction of the presheaf $G \to \mathbf{Grpd}$ that the action constitutes; it is a special case of a homotopy colimit. (The notation $X /\!/ G$ is often used [9].) If $G$ acts on the point groupoid $1$, then $1/G$ is the groupoid with one object and vertex group $G$.

If $p : X \to B$ is a morphism of groupoids, for $b \in B$ the 'inclusion' of the homotopy fibre $X_b \to X$ is faithful but not full in general. But $\mathrm{Aut}(b)$ acts on $X_b$ canonically, and the homotopy quotient $X_b / \mathrm{Aut}(b)$ provides exactly the missing arrows, so as to make the natural map $X_b / \mathrm{Aut}(b) \to X$ fully faithful.

Since every object $x \in X$ must map to some connected component of $B$, we find the equivalence

$$X \simeq \sum_{b \in \pi_0 B} X_b / \operatorname{Aut}(b) =: \int^{b \in B} X_b, \qquad (9)$$

expressing $X$ as the homotopy sum of the fibres, or equivalently as a family of groupoids (indexed by $\pi_0(B)$ and with a group action in each). Given morphisms of groupoids $Y \xrightarrow{p} B \xrightarrow{f} A$, we have the following 'Fubini formula':

$$\int^{b \in B} Y_b \simeq \int^{a \in A} \int^{b \in B_a} Y_b$$

which is actually the formula for the 'dependent-sum' functor $\Sigma_f : \mathbf{Grpd}_{/B} \to \mathbf{Grpd}_{/A}$ given by postcomposition. In family notation the formula reads

$$\Sigma_f(Y_b \mid b \in B) = (\textstyle\int^{b \in B_a} Y_b \mid a \in A),$$

just as Formula (6) in the **Set** case.

Pullback along $f : B \to A$, denoted $\Delta_f$, is right adjoint to $\Sigma_f$. This means of course homotopy adjoint, and amounts to a natural equivalence of mapping groupoids $\mathbf{Grpd}_{/A}(\Sigma_f Y, X) \simeq \mathbf{Grpd}_{/B}(Y, \Delta_f X)$. The proof relies on the universal property of the pullback. One may note the following formula for pullback, in family notation:

$$\Delta_f(X_a \mid a \in A) = (X_{f(b)} \mid b \in B),$$

again completely analogous to the **Set** case (Formula (5)).

The 2-category of groupoids is locally cartesian closed. This means that the pullback functor in turn has a right adjoint $\Pi_f : \mathbf{Grpd}_{/B} \to \mathbf{Grpd}_{/A}$. The general formula is an end formula; for $Y \to B$, the fibre of $\Pi_f Y$ over $a \in A$ can be described explicitly as the mapping groupoid

$$(\Pi_f Y)_a = \mathbf{Grpd}_{/B}(B_a, Y).$$

(A more explicit formula will be derived in the discrete case below.)

**2.4 Polynomial functors.** A *polynomial* is a diagram of groupoids

$$I \xleftarrow{s} E \xrightarrow{p} B \xrightarrow{t} J.$$

The associated *polynomial functor* (or the *extension* of the polynomial) is given as the composite

$$\mathbf{Grpd}_{/I} \xrightarrow{\Delta_s} \mathbf{Grpd}_{/E} \xrightarrow{\Pi_p} \mathbf{Grpd}_{/B} \xrightarrow{\Sigma_t} \mathbf{Grpd}_{/J}.$$

**2.5 Beck–Chevalley, distributivity, and composition.** Given a homotopy pullback square

$$
\begin{array}{ccc}
\cdot & \xrightarrow{\ \psi\ } & \cdot \\
{\scriptstyle\alpha}\downarrow & {\scriptstyle\lrcorner} & \downarrow{\scriptstyle\beta} \\
\cdot & \xrightarrow{\ \varphi\ } & \cdot
\end{array}
$$

there are natural equivalences of functors

$$
\Sigma_\alpha \circ \Delta_\psi \overset{\sim}{\Rightarrow} \Delta_\varphi \circ \Sigma_\beta \qquad \text{and} \qquad \Delta_\beta \circ \Pi_\varphi \overset{\sim}{\Rightarrow} \Pi_\psi \circ \Delta_\alpha,
$$

usually called the Beck–Chevalley conditions. A more subtle feature of the theory is distributivity, which in this setting is an equivalence saying how to distribute dependent products over dependent sums (and which can be interpreted as a type-theoretic form of the axiom of choice [39]). We shall not need the details here. See [18] for the classical case, and Weber [46] for a deeper treatment. The Beck–Chevalley conditions and distributivity yield a formula for composing polynomial functors [18].

**2.6 Natural transformations.** Just as in the classical case [18], homotopy cartesian natural transformations $P' \Rightarrow P$ of polynomial functors (in one variable) correspond precisely to homotopy cartesian diagrams

$$
\begin{array}{ccc}
E' & \longrightarrow & B' \\
\downarrow & {\scriptstyle\lrcorner} & \downarrow \\
E & \longrightarrow & B.
\end{array}
$$

This is an easy consequence of Beck–Chevalley. Showing more generally that arbitrary natural transformations are given essentially uniquely by diagrams

$$
\begin{array}{ccc}
E' & \longrightarrow & B' \\
\uparrow & & \| \\
\cdot & \longrightarrow & B' \\
\downarrow & {\scriptstyle\lrcorner} & \downarrow \\
E & \longrightarrow & B
\end{array}
$$

is a bit more involved and depends on a homotopy version of the Yoneda lemma. (At the time of this writing, this result is not as precise as in the 1-dimensional case of [18].)

**2.7 Spans and stuff operators.** Spans of groupoids are the special case of groupoid polynomials where the middle map is the identity (or an equivalence). These constitute a categorification of matrix algebra, and were called *stuff operators* by Baez and Dolan [9]; they have been used to give groupoid models for certain aspects of Hecke algebras and Hall algebras [10].

# 3   Exactness; combinatorial polynomial functors

The following results from [21] are actually proved in the much richer setting of $\infty$-groupoids, but the proofs work also for 1-groupoids. We now suppress the clumsy 'homotopy' everywhere, although of course all limits and colimits mentioned refer to the homotopy notions.

**Theorem 3.1** *(Gepner-Kock [21].)   A functor $\mathbf{Grpd}_{/I} \to \mathbf{Grpd}_{/J}$ is polynomial if and only if it is accessible and preserves conical limits.*

By *conical limit* we mean limit over a diagram with a terminal vertex. Recall that a functor is accessible [38, Ch. 5] when it preserves $\kappa$-filtered colimits for some regular cardinal $\kappa$. The regular cardinal here is explicitly characterised:

**Proposition 3.2** *([21])   A polynomial functor given by $I \leftarrow E \xrightarrow{p} E \to J$ preserves $\kappa$-filtered colimits if and only if $p$ has $\kappa$-compact fibres.*

An important case is $\kappa = \omega$. A groupoid is $\omega$-compact when it has finitely many components (i.e. $\pi_0(X)$ is a finite set) and all vertex groups are finitely presented.

**3.3 Discreteness.**   For many data types occurring in practice (including species and all the examples below), although they may have symmetries, the positions in each shape form a *discrete* groupoid, i.e. a groupoid equivalent to a set. In the polynomial formalism this amounts to the middle map $p : E \to B$ having discrete fibres. In this case, the dependent product formula simplifies to

$$(\Pi_p Y)_b = \prod_{e \in \pi_0(E_b)} Y_e,$$

in analogy with (7), and hence all the formulae look exactly like the **Set** case.

The corresponding exactness condition is preservation of sifted colimits. A $\kappa$-*sifted colimit* is a colimit over a diagram $D$ whose diagonal $D \to D^S$ is cofinal for every set $S$ of cardinality $< \kappa$ [38, Ch. 5].

**Proposition 3.4** *([21])   A polynomial functor given by $I \leftarrow E \xrightarrow{p} E \to J$ preserves $\kappa$-sifted colimits if and only if $p$ has $\kappa$-compact discrete fibres.*

**3.5 Combinatorial polynomial functors.**   We call a polynomial functor $I \leftarrow E \xrightarrow{p} B \to J$ *combinatorial* if the fibres of $p$ are equivalent to finite sets (i.e. are $\omega$-compact discrete).

**3.6 Species in groupoids (stuff types).**   A Baez-Dolan *stuff type* [9] is a map of groupoids $F \to \mathbb{B}_\omega$. We prefer the name *species in groupoids*. (A classical species is when the map has discrete fibres, or equivalently is faithful.)

Its extension is the left homotopy Kan extension of $n \mapsto F_n$ along $\mathbb{B}_\omega \subset \textbf{Grpd}$:

$$\begin{array}{rcl} \textbf{Grpd} & \longrightarrow & \textbf{Grpd} \\ X & \longmapsto & \displaystyle\sum_{n \in \pi_0(\mathbb{B}_\omega) = \mathbb{N}} \frac{F_n \times X^n}{\text{Aut}(n)} \, . \end{array}$$

(That's a homotopy quotient of course.)

This functor is polynomial [21]: the representing groupoid map is the top row in the pullback

$$\begin{array}{ccc} E & \longrightarrow & F \\ \downarrow & \lrcorner & \downarrow \\ \mathbb{B}'_\omega & \longrightarrow & \mathbb{B}_\omega. \end{array}$$

This map has finite discrete fibres since $\mathbb{B}'_\omega \to \mathbb{B}_\omega$ has. (Here $\mathbb{B}'_\omega$ is the groupoid of finite pointed sets.) Conversely, given a groupoid polynomial $E \to F$ with finite discrete fibres, the 'classifying map' $F \to \mathbb{B}_\omega$ (obtained since $\mathbb{B}'_\omega \to \mathbb{B}_\omega$ classifies finite discrete fibrations) yields a species in groupoids. One can check that the extension of the polynomial agrees with the extension of the species. In conclusion:

**Proposition 3.7** *([21]) Combinatorial polynomial functors $\textbf{Grpd} \to \textbf{Grpd}$ are the same thing as analytic functors (in the sense of Baez-Dolan).*

Combining these results we get a 'Joyal theorem':

**Corollary 3.8** *([21]) A functor $\textbf{Grpd} \to \textbf{Grpd}$ is analytic (in the sense of Baez-Dolan) if and only if it preserves $\omega$-sifted colimits and conical limits.*

**3.9 Generalised species.** The relationship between polynomial functors and the generalised species of [15] has been sketched by Gambino and the author (unpublished). A generalised species depends on two categories $I$ and $J$, and has as extension a generalised analytic functor $\textbf{PrSh}(I) \to \textbf{PrSh}(J)$; this generalises the 1985 notion but not the Baez-Dolan notion. If $I$ and $J$ are groupoids, these generalised analytic functors correspond to the *'classical'* extension of combinatorial polynomials over groupoids, i.e. involving $\pi_0$ on quotients.

**3.10 Examples.** Groupoid polynomials encode data types in groupoids. For example, $\mathbb{B}'_\omega \to \mathbb{B}_\omega$ encodes the multiset data type: the groupoid $\mathbb{B}_\omega$ of finite sets and bijections is the groupoid of shapes — the shape of a multiset is really the set indexing its elements, not just its size. There are $\mathbb{N}$-many isoclasses; the isomorphisms should be interpreted as propositional equality. The fibre over $S \in \mathbb{B}_\omega$ is the discrete groupoid of positions in $S$, i.e. a uniform prescription of positions in multisets indexed by $S$. Indeed, since $\mathbb{B}'_\omega \to \mathbb{B}_\omega$ is a fibration, the fibre is canonically identified with the set $S$ itself — note its natural $\text{Aut}(S)$-action. The discreteness of the fibre means that propositional

equality between positions can be regarded as definitional equality. The extension of this quotient container is naturally an endofunctor $\mathbf{Grpd} \to \mathbf{Grpd}$. But one obtains an endofunctor $\mathbf{Set} \to \mathbf{Set}$ (in this case precisely (8)) by precomposing with the natural inclusion $\mathbf{Set} \to \mathbf{Grpd}$ and postcomposing with $\pi_0 : \mathbf{Grpd} \to \mathbf{Set}$. The first is harmless. The second corresponds to collapsing all isomorphisms to identity, i.e. interpreting propositional equality as definitional equality. If the argument is a set, the only collapse is the passage from homotopy quotient to naïve quotient (of actions on sets).

The data type of cyclic lists is groupoid polynomial, represented by $\mathbb{C}'_\omega \to \mathbb{C}_\omega$, where $\mathbb{C}_\omega$ is the groupoid of finite cyclically ordered sets, and $\mathbb{C}'_\omega$ is the groupoid of pointed cyclically ordered finite sets. From 1.1, the list data type is represented by $\mathbb{N}' \to \mathbb{N}$, interpreted as the groupoids of linearly ordered finite sets and pointed ditto. The diagram of groupoids

$$
\begin{array}{ccc}
\mathbb{N}' & \longrightarrow & \mathbb{N} \\
\downarrow & \lrcorner & \downarrow \\
\mathbb{C}'_\omega & \longrightarrow & \mathbb{C}_\omega \\
\downarrow & \lrcorner & \downarrow \\
\mathbb{B}'_\omega & \longrightarrow & \mathbb{B}_\omega
\end{array}
$$

now represents the cartesian natural transformations, or polymorphic functions, from lists to cyclic lists to multisets.

## 4 Trees

W-types in Martin-Löf type theory correspond to initial algebras of polynomial functors (cf. [40] and [17] for the extensional case, and [8] for the fully intensional case). The initial algebra for $1 + P$ can also be described as the set of operations for the free monad on $P$, which in turn is the set of $P$-*trees*. $P$-trees (for $P$ a polynomial functor over $\mathbf{Set}$ or any lccc) are always rigid, i.e. have no symmetries. Abstract trees, on the other hand, admit symmetries, so they are not $P$-trees for any $\mathbf{Set}$-polynomial functor $P$, and they are neither W-types nor containers in the classical sense. Instead, according to [32], abstract trees *are* themselves polynomial functors. It is convenient to take the following characterisation of trees as a definition:

**4.1 Trees.** ([32]) A *(finite) tree* is a diagram of finite sets

$$
A \xleftarrow{s} M \xrightarrow{p} N \xrightarrow{t} A
$$

satisfying the following three conditions:

(1) $t$ is injective

(2) $s$ is injective with singleton complement (called *root* and denoted 1). With $A = 1 + M$, define the walk-to-the-root function $\sigma : A \to A$ by $1 \mapsto 1$ and $e \mapsto t(p(e))$ for $e \in M$.

(3) $\forall x \in A : \exists k \in \mathbb{N} : \sigma^k(x) = 1$.

The elements of $A$ are called *edges*. The elements of $N$ are called *nodes*. For $b \in N$, the edge $t(b)$ is called the *output edge* of the node. That $t$ is injective is just to say that each edge is the output edge of at most one node. For $b \in N$, the elements of the fibre $M_b$ are called *input edges* of $b$. Hence the whole set $M = \sum_{b \in N} M_b$ can be thought of as the set of nodes-with-a-marked-input-edge, i.e. pairs $(b, e)$ where $b$ is a node and $e$ is an input edge of $b$. The map $s$ returns the marked edge. Condition (2) says that every edge is the input edge of a unique node, except the root edge. Condition (3) says that if you walk towards the root, in a finite number of steps you arrive there. The edges not in the image of $t$ are called *leaves*.

**4.2 Decorated trees: $P$-trees** ([32]; see also [33, 34, 35]) An efficient way of encoding and manipulating decorations of trees is in terms of polynomial endofunctors. Let $P$ be a polynomial endofunctor given by $I \xleftarrow{d} E \xrightarrow{q} B \xrightarrow{c} I$. A $P$-*tree* is a diagram

$$
\begin{array}{ccccccc}
A & \longleftarrow & M & \longrightarrow & N & \longrightarrow & A \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
I & \longleftarrow & E & \longrightarrow & B & \longrightarrow & I \ ,
\end{array}
\tag{10}
$$

where the top row is a tree. The squares are commutative up to isomorphism, and it is important that the 2-cells be specified as part of the structure. Unfolding the definition, we see that a $P$-tree is a tree whose edges are decorated in $I$, whose nodes are decorated in $B$, and with the additional structure of an equivalence $M_n \simeq E_b$ for each node $n \in N$ with decoration $b \in B$ (this is essentially just a bijection, since the fibres are discrete), an iso in $I$ between the decoration of an edge $m \in M_n$ and the corresponding $d(e)$, and finally an iso in $I$ between the decoration of the output edge of $n$ and $c(b)$.

**4.3 Examples of $P$-trees.** Natural numbers are $P$-trees for the identity monad $P(X) = X$, and are also the set of operations of the list monad. Planar finite trees are $P$-trees for $P$ the list monad, and are also the set of operations of the free-non-symmetric-operad monad [37]. These two examples are the first entries of a canonical sequence of inductive data types underlying several approaches to higher category theory, the *opetopes*: opetopes in dimension $n$ are $P$-trees for $P$ a **Set**-polynomial functor whose operations are $(n-1)$-opetopes [35]; hence opetopes are higher-dimensional trees.

Abstract finite trees are $P$-trees for the multiset functor $1 \leftarrow \mathbb{B}'_\omega \to \mathbb{B}_\omega \to 1$, but cannot be realised as $P$-trees for any **Set**-polynomial $P$.

**4.4 Trees of Feynman graphs.** In the so-called BPHZ renormalisation of perturbative quantum field theories, one is concerned with nestings of 1-particle irreducible (1PI) Feynman graphs, i.e. graphs [30] for which no single edge removal disconnects. Kreimer [36] discovered that the BPHZ procedure is encoded in a Hopf algebra of (non-planar) rooted trees, expressing the nesting of graphs.
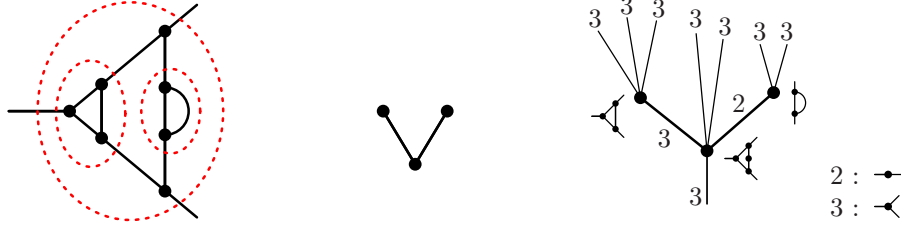


In the picture the combinatorial tree in the middle expresses the nesting of 1PI subgraphs on the left; such trees are sufficient in Kreimer's Hopf-algebra approach to BPHZ, but do not capture the symmetries of the graph. To this end, further decoration is needed in the tree, as partially indicated on the right. First of all, each node in the tree should be decorated by the 1PI graph it corresponds to in the nesting, and second, the tree should have leaves (input slots) corresponding to the vertices of the graph. The decorated tree should be regarded as a recipe for reconstructing the graph by inserting the decorating graphs into the vertices of the graphs of parent nodes. The numbers on the edges indicate the type constraint of each substitution: the outer interface of a graph must match the local interface of the vertex it is substituted into. But the type constraints on the tree decoration are not enough to reconstruct the graph, because for example the small graph ⤙ decorating the left-hand node could be substituted into various different vertices of the graph ⤙.

The solution found in [34] is to consider $P$-trees, for $P$ the polynomial endofunctor given by $I \xleftarrow{s} E \xrightarrow{p} B \xrightarrow{t} I$, where $I$ is the groupoid of interaction labels for the theory (in this case the one-vertex graphs ⤚ and ⤙ ) and $B$ is the groupoid of connected 1PI graphs of the theory, and $E$ is the groupoid of such 1PI graphs with a marked vertex. The map $s$ returns the one-vertex subgraph at the mark, $p$ forgets the mark, and $t$ returns the outer interface of the graph, i.e. the graph obtained by contracting everything to a point, but keeping the external lines. A $P$-tree is hence a diagram like (10) with specified 2-cells. These 2-cells carry much of the structure: for example the 2-cell on the right says that the 1PI graph decorating a given node must have the same outline as the decoration of the outgoing edge of the node — or more precisely, and more realistically: an isomorphism is specified (it's a bijection between external lines of one-vertex graphs). Similarly, the left-hand 2-cell specifies for each node-with-a-marked-incoming-edge $x' \in M$, an isomorphism between the one-vertex graph decorating that edge and the marked vertex of the graph

13

decorating the marked node $x'$. Hence the structure of a $P$-tree is a complete recipe not only for which graphs should be substituted into which vertices, but also *how*: specific bijections prescribe which external lines should be identified with which lines in the receiving graph. In fact, *there is an equivalence of groupoids between nested graphs and P-trees* [34]. This is exploited in [16] to establish algebraic identities concerning graphs by interpreting them as homotopy cardinalities of equivalences of groupoids of decorated trees.

Notice that the polynomial functor $P$ is combinatorial, since each graph has a discrete finite set of vertices. It is not a species in the classical sense though: the classifying map $B \to \mathbb{B}_\omega$ sends a graph to its set of vertices, and since a graph may have nontrivial automorphisms that fix every vertex, this map does not have discrete fibres.

## 5    Outlook

A 2-category is called *locally cartesian closed* when for every arrow $f : B \to A$, we have the string of adjoint functors $\Sigma_f \dashv \Delta_f \dashv \Pi_f$. This structure formally implies the Beck-Chevalley equivalences and distributivity, which are the minimal requirements for a reasonable theory of polynomial functors. The theory of strength can be copied almost verbatim from [18], and it seems that the representation theorem of [18] also carries over.

While locally cartesian closed categories provides semantics for an extensional version of Martin-Löf type theory [44], [13], and locally cartesian closed 2-categories capture some 2-truncated version ([25], [19]), recent insight of Homotopy Type Theory strongly suggests that in the long run, the case of $\infty$-groupoids and other locally cartesian closed $\infty$-categories will be the real meat for type theory. Large parts of the $\infty$-theory of polynomial functors, as well as aspects of the theory of locally cartesian closed $\infty$-categories geared towards Homotopy Type Theory have already been worked out in joint work with David Gepner, and will appear elsewhere [21], [20]. Nevertheless the groupoid case is interesting in its own right, since it already covers important applications: in particular for many purposes of combinatorial nature, 1-groupoids are all that is needed in order to handle symmetry issues. Time will tell whether for purposes of program semantics the groupoid level is enough too — otherwise it is a good stepping stone into the $\infty$-world.

## References

[1]  Abbott, M., "Categories of Containers," Ph.D. thesis, University of Leicester, 2003. Available from http://www.mcs.le.ac.uk/~ma139/docs/thesis.pdf.

[2]  Abbott, M., T. Altenkirch and N. Ghani, *Categories of containers*, in: *Foundations of software science and computation structures*, Lecture Notes in Comput. Sci. **2620** (2003), pp. 23–38.

[3] Abbott, M., T. Altenkirch and N. Ghani, *Containers: constructing strictly positive types*, Theoret. Comput. Sci. **342** (2005), pp. 3–27.

[4] Abbott, M., T. Altenkirch, N. Ghani and C. McBride, *Derivatives of containers*, in: *Typed lambda calculi and applications (Valencia, 2003)*, Lecture Notes in Comput. Sci. **2701** (2003), pp. 16–30.

[5] Abbott, M., T. Altenkirch, N. Ghani and C. McBride, *Constructing polymorphic programs with quotient types*, in: *Mathematics of program construction*, Lecture Notes in Comput. Sci. **3125** (2004), pp. 2–15.

[6] Adámek, J. and J. Velebil, *Analytic functors and weak pullbacks*, Theory Appl. Categ. **21** (2008), pp. 191–209.

[7] Altenkirch, T. and P. Morris, *Indexed containers*, in: *Proceedings of the Twenty-Fourth Annual IEEE Symposium on Logic in Computer Science (LICS 2009)* (2009), pp. 277–285.

[8] Awodey, S., N. Gambino and K. Sojakova, *Inductive types in homotopy type theory*, preprint, ArXiv:1201.3898.

[9] Baez, J. C. and J. Dolan, *From finite sets to Feynman diagrams*, in: B. Engquist and W. Schmid, editors, *Mathematics unlimited—2001 and beyond* (2001), pp. 29–50. ArXiv:math.QA/0004133.

[10] Baez, J. C., A. E. Hoffnung and C. D. Walker, *Higher dimensional algebra VII: groupoidification*, Theory Appl. Categ. **24** (2010), pp. 489–553. ArXiv:0908.4305.

[11] Bergeron, F., G. Labelle and P. Leroux, "Combinatorial species and tree-like structures," Encyclopedia of Mathematics and its Applications **67**, Cambridge University Press, Cambridge, 1998, xx+457 pp.

[12] Carette, J. and G. Uszkay, *Species: making analytic functors practical for functional programming* (2008), available from http://www.cas.mcmaster.ca/~carette/species/.

[13] Clairambault, P. and P. Dybjer, *The biequivalence of locally cartesian closed categories and Martin-Löf type theories*, preprint, ArXiv:1112.3456.

[14] Fiore, M., *Mathematical models of computational and combinatorial structures (invited address)*, in: *Foundations of software science and computation structures*, Lecture Notes in Comput. Sci. **3441** (2005), pp. 25–46.

[15] Fiore, M., N. Gambino, M. Hyland and G. Winskel, *The Cartesian closed bicategory of generalised species of structures*, J. Lond. Math. Soc. (2) **77** (2008), pp. 203–220.

[16] Gálvez-Carrillo, I., J. Kock and A. Tonks, *Groupoids and Faà di Bruno formulae for Green functions in bialgebras of trees*, preprint, ArXiv:1207.6404.

[17] Gambino, N. and M. Hyland, *Wellfounded trees and dependent polynomial functors*, in: S. Berardi, M. Coppo and F. Damiani, editors, *TYPES 2003*, Lecture Notes in Comput. Sci. **3085** (2004), pp. 210–225.

[18] Gambino, N. and J. Kock, *Polynomial functors and polynomial monads*, to appear in Math. Proc. Cambridge Philos. Soc., ArXiv:0906.4931.

[19] Garner, R., *Two-dimensional models of type theory*, Math. Struct. Comput. Sci. **19** (2009), pp. 687–736. ArXiv:0808.2122.

[20] Gepner, D. and J. Kock, *Univalence in locally cartesian closed ∞-categories*, preprint, ArXiv:1208.1749.

[21] Gepner, D. and J. Kock, *Polynomial functors over infinity categories*, in preparation.

[22] Gylterud, H., *Symmetric Containers*, Master's thesis, University of Oslo, 2011. Available from http://www.duo.uio.no/publ/matematikk/2011/144617/thesisgylterud.pdf.

[23] Hancock, P. and A. Setzer, *Interactive programs in dependent type theory*, in: *Computer science logic (Fischbachau, 2000)*, Lecture Notes in Comput. Sci. **1862** (2000), pp. 317–331,

[24] Hasegawa, R., *Two applications of analytic functors*, Theoret. Comput. Sci. **272** (2002), pp. 113–175.

[25] Hofmann, M. and T. Streicher, *The groupoid interpretation of type theory*, in: *Twenty-five years of constructive type theory*, Oxford Logic Guides **36**, Oxford University Press, 1998, pp. 83–111.

[26] Joyal, A., *Une théorie combinatoire des séries formelles*, Adv. Math. **42** (1981), pp. 1–82.

[27] Joyal, A., *Foncteurs analytiques et espèces de structures*, in: *Combinatoire énumérative (Montréal/Québec, 1985)*, Lecture Notes in Mathematics **1234**, Springer, 1986, pp. 126–159.

[28] Joyal, A., *Quasi-categories and Kan complexes*, J. Pure Appl. Alg. **175** (2002), pp. 207–222.

[29] Joyal, A., "The theory of quasi-categories," Advanced Course on Simplicial Methods in Higher Categories, vol. II, Quaderns, núm. 45 (2008), pp. 147–497, CRM Barcelona. Available at http://mat.uab.cat/~kock/crm/hocat/advanced-course/Quadern45-2.pdf.

[30] Joyal, A. and J. Kock, *Feynman graphs, and nerve theorem for compact symmetric multicategories (extended abstract)*, in: *Proceedings of the 6th International Workshop on Quantum Physics and Logic (Oxford 2009)*, Elec. Notes Theoret. Comput. Sci. **270** (2011), pp. 105–113. ArXiv:0908.2675.

[31] Kock, A. and J. Kock, *Local fibered right adjoints are polynomial*, to appear in Math. Struct. Comput. Sci., ArXiv:1005.4236.

[32] Kock, J., *Polynomial functors and trees*, Internat. Math. Res. Notices **2011** (2011), pp. 609–673. ArXiv:0807.2874.

[33] Kock, J., *Categorification of Hopf algebras of rooted trees*, to appear in Centr. Eur. J. Math., ArXiv:1109.5785.

[34] Kock, J., *Categorical formalisms for graphs and trees in quantum field theory* (in preparation).

[35] Kock, J., A. Joyal, M. Batanin and J.-F. Mascari, *Polynomial functors and opetopes*, Adv. Math. **224** (2010), pp. 2690–2737. ArXiv:0706.1033.

[36] Kreimer, D., *On the Hopf algebra structure of perturbative quantum field theories*, Adv. Theor. Math. Phys. **2** (1998), pp. 303–334. ArXiv:q-alg/9707029.

[37] Leinster, T., "Higher Operads, Higher Categories," London Math. Soc. Lecture Note Series, Cambridge University Press, Cambridge, 2004, 448 pp. ArXiv:math.CT/0305049.

[38] Lurie, J., "Higher topos theory," Annals of Mathematics Studies **170**, Princeton University Press, Princeton, NJ, 2009, xviii+925 pp. ArXiv:math/0608040.

[39] Martin-Löf, P., "Intuitionistic Type Theory," Bibliopolis, Napoli, 1984, iv+91 pp.

[40] Moerdijk, I. and E. Palmgren, *Wellfounded trees in categories*, Ann. Pure Appl. Logic **104** (2000), pp. 189–218.

[41] Nordström, B., K. Petersson and J. M. Smith, "Programming in Martin-Löf's type theory, an introduction." International Series of Monographs on Computer Science **7**, The Clarendon Press Oxford University Press, New York, 1990, x+221 pp.

[42] Petersson, K. and D. Synek, *A set constructor for inductive sets in Martin-Löf type theory*, in: *Category Theoy and Computer Science (Manchester 1989)*, Lecture Notes in Comput. Sci. **389** (1989), pp. 128–140.

[43] Rutten, J. J. M. M., *Universal coalgebra: a theory of systems*, Theoret. Comput. Sci. **249** (2000), pp. 3–80.

[44] Seely, R. A. G., *Locally cartesian closed categories and type theory*, Math. Proc. Cambridge Phil. Soc. **95** (1984), pp. 33–48.

[45] Tambara, D., *On multiplicative transfer*, Comm. Alg. **21** (1993), pp. 1393–1420.

[46] Weber, M., *Polynomials in categories with pullbacks*, preprint, ArXiv:1106.1983.

[47] Yorgey, B. A., *Species and functors and types, oh my!*, in: *Proceedings of the third ACM Haskell symposium on Haskell*, Haskell '10 (2010), pp. 147–158, available from http://www.cis.upenn.edu/~byorgey/pub/species-pearl.pdf