Towards a Functional Language for Species of Structures

Zeinab Galal
IRIF, University Paris Diderot
France
zgalal@irif.fr

1 Introduction

Species of structures were first introduced by Joyal as a unified framework for the theory of generating series in enumerative combinatorics. In 1988, Girard introduced normal functors as a model of pure λ-calculus where terms are interpreted as infinite series with sets as coefficients (which correspond to a special case of Joyal's species). Fiore presented a generalized definition that both encompasses Joyal's species and constitutes a model of differential linear logic.

Since species encode families of labelled structures, much work has been done to investigate their connection with algebraic data types (see [8], [2] and [7]). We want to explore an alternative viewpoint of seeing them as terms in an extension of λ-calculus motivated by the relationship between species and differential models of linear logic and use the combinatorial intuition as a guide in the design of the syntax. The next step would be to study methods of resolution of differential equations in the setting of generalized species with the ultimate goal being to establish in linear logic a combinatorial interpretation of the obtained differential calculus.

2 Species of Structures

2.1 Combinatorial Species

A combinatorial species is a class of structures on arbitrary finite sets of labels which is invariant under relabellings along bijections. Formally, a species is a functor $\mathbb{F} : \mathbb{B} \to \mathbf{Set}$ from the category of finite sets and bijections $\mathbb{B}$ to the category **Set**. The functoriality of $\mathbb{F}$ means that labels do not matter. Species can be added together, multiplied, differentiated, and substituted one into another to form new species from given ones.

Species	One	Singleton	Lists	Cycles	Sets
Definition	$\mathbb{1} : \mathbb{B} \to \mathbf{Set}$ $\mathscr{U} \mapsto \begin{cases} \{\star\} \text{ if } \mathscr{U} = \emptyset \\ \emptyset \text{ otherwise} \end{cases}$	$\mathbb{X} : \mathbb{B} \to \mathbf{Set}$ $\mathscr{U} \mapsto \begin{cases} \mathscr{U} \text{ if } \lvert\mathscr{U}\rvert = 1 \\ \emptyset \text{ otherwise} \end{cases}$	$\mathbb{L} : \mathbb{B} \to \mathbf{Set}$ $\mathscr{U} \mapsto \mathrm{Bij}(\mathscr{U}, \mathscr{U})$	$\mathbb{C} : \mathbb{B} \to \mathbf{Set}$ $\mathscr{U} \mapsto \{f : \mathscr{U} \xrightarrow{\sim} \mathscr{U} \mid f \text{ is a cycle}\}$	$\mathbb{E} : \mathbb{B} \to \mathbf{Set}$ $\mathscr{U} \mapsto \{\mathscr{U}\}$
Generating series	$x \mapsto 1$	$x \mapsto x$	$x \mapsto \sum_{n\in\mathbb{N}} x^n$	$x \mapsto \sum_{n\in\mathbb{N}} \frac{x^n}{n}$	$x \mapsto \sum_{n\in\mathbb{N}} \frac{x^n}{n!}$

Figure 1: Examples of species

Definition 2.1. A species $\mathbb{F}$ is *molecular* if for all species $\mathbb{G}$ and $\mathbb{H}$, $\mathbb{F} = \mathbb{G} + \mathbb{H}$ implies $\mathbb{G} = \mathbb{0}$ or $\mathbb{H} = \mathbb{0}$.

Submitted to:
Linearity and TLLA 2018

Every species has a unique decomposition into a sum of molecular species and it is on the level of molecular species that we construct integrals or solutions to differential equations in the combinatorial setting, it is hence crucial to first generalize these notions to our context.

Theorem 2.2 (see [1])**.** *Given a species $\mathbb{F} : \mathbb{B} \to \mathbf{Set}$, the following are equivalent:*

1. *$\mathbb{F}$ is molecular*
2. *There exist an $n \in \mathbb{N}$ and H a subgroup of $\mathfrak{S}_n$ such that $\mathbb{F}$ is isomophic to the species:*

$$\mathscr{U} \in \mathbb{B} \mapsto Bij(\underline{n}, \mathscr{U})/H \quad \text{where } \underline{n} := \{0, 1, \dots, n-1\}$$

3. *$\mathbb{F}$ is transitive, i.e. for all $n, n' \in \mathbb{N}$, $x \in \mathbb{F}[\underline{n}]$ and $x' \in \mathbb{F}[\underline{n'}]$, there exists $f : \underline{n} \to \underline{n'}$ such that $\mathbb{F}[f](x) = x'$.*

If we restrict ourselves to species constituted only of molecular species where we quotient by the trivial subgroup H, we obtain species that are called *flat* ant they are equivalent to the normal functor model defined by Girard for which we have a notion of integration.

2.2 Generalized Species

In [4], Fiore et al. defined a more general notion of species as 1-cells in the co-Kleisli of the bicategory of profunctors **Prof** based on the fact that a functor $\mathbb{B} \to \mathbf{Set}$ is equivalent to a functor $!\mathbb{1} \to \widehat{\mathbb{1}}$ where $!\mathbb{1}$ is the free symmetric strict monoidal completion of the category $\mathbb{1}$ and $\widehat{\mathbb{1}}$ is the category of presheaves on $\mathbb{1}$.

Definition 2.3. Given $\mathbf{A}$ and $\mathbf{B}$ two small categories, an $(\mathbf{A}, \mathbf{B})$-*generalized species of structures* is a profunctor $\mathbb{F} : !\mathbf{A} \nrightarrow \mathbf{B}$ (or equivalently a functor $\mathbb{F} : !\mathbf{A} \times \mathbf{B}^{op} \to \mathbf{Set}$).

They also showed that it was also possible to define operations of addition, multiplication and composition compatible with the previous definitions on combinatorial species and that generalized species are a model of differential linear logic. We want to interpret our extension of the simply typed λ-calculus as generalized species in the bicategory of profunctors. In this setting, species are viewed as terms and all the operations on species are term operations rather than operations on types.

3 Syntax

Types $\sigma, \tau ::= \bot \mid \sigma \to \tau$ (as a first step to tackle the general problem, we will only work with the set of simple types generated by a unique atomic type $\bot$)

Terms $t, s ::= \star \mid x \mid (t)s \mid \lambda x^{\sigma}.t \mid t + s \mid t \cdot s \mid \mathbf{fix}\, x^{\sigma} \cdot t \mid (\mathbf{D}t)s$

$$\frac{}{\Gamma, x : \sigma \vdash x : \sigma}\ \text{AX} \qquad \frac{}{\Gamma \vdash \star : \bot}\ \bot \qquad \frac{\Gamma \vdash t : \sigma \quad \Gamma \vdash s : \sigma}{\Gamma \vdash t + s : \sigma}\ \text{SUM} \qquad \frac{\Gamma \vdash t : \sigma \quad \Gamma \vdash s : \sigma}{\Gamma \vdash t \cdot s : \sigma}\ \text{MULT} \qquad \frac{\Gamma, x : \sigma \vdash t : \tau}{\Gamma \vdash \lambda x^{\sigma}.t : \sigma \to \tau}\ \text{LAM}$$

$$\frac{\Gamma \vdash t : \sigma \to \tau \quad \Gamma \vdash s : \sigma}{\Gamma \vdash (t)s : \tau}\ \text{APP} \qquad \frac{\Gamma \vdash t : \sigma \to \sigma}{\Gamma \vdash \mathbf{fix}(t) : \sigma}\ \text{FIX} \qquad \frac{\Gamma \vdash t : \sigma \to \tau \quad \Gamma \vdash s : \sigma}{\Gamma \vdash (\mathbf{D}t)s : \sigma \to \tau}\ \text{DIFF}$$

Figure 2: Typing rules

4 Semantics

Interpretation of types To any type σ, we associate an object $[\![\sigma]\!]$ in **Prof** as follows: $[\![\bot]\!] := \mathbb{1}$ and $[\![\sigma \to \tau]\!] := ![\![\sigma]\!]^{op} \times [\![\tau]\!]$. To a typing context $\Gamma = (x_1 : \sigma_1, \dots, x_n : \sigma_n)$, we set $[\![\Gamma]\!] := ![\![\sigma_1]\!] \otimes \cdots \otimes ![\![\sigma_n]\!]$.

$$\frac{}{(\lambda x^{\sigma}.t)s \to t[s/x]} \qquad \frac{t \to t'}{(t)s \to (t')s} \qquad \frac{s \to s'}{(t)s \to (t)s'} \qquad \frac{t \to t'}{t+s \to t'+s} \qquad \frac{t \to t'}{t\cdot s \to t'\cdot s} \qquad \frac{}{\mathbf{fix}(t) \to (t)\,\mathbf{fix}(t)}$$

Figure 3: Some reduction rules

Intepretation of typing rules Given a typing judgment $\Gamma \vdash t : \tau$, we define inductively a profunctor $[\![t]\!]_\Gamma : ![\![\Gamma]\!] \nrightarrow [\![\tau]\!]$. We will only present a couple of examples for the interpretation due to lack of space:

Variable: $[\![x]\!]_{\Gamma,x:\sigma} : ![\![\Gamma]\!] \times ![\![\sigma]\!] \nrightarrow [\![\sigma]\!]$

$$((u,v),a) \mapsto \mathbf{Hom}_{![\![\sigma]\!]}(\langle a\rangle, v)$$

This profunctor will enable us to express a generalized version of the singleton species $\mathbb{X}$

Bottom: $[\![\star]\!]_\Gamma : ![\![\Gamma]\!] \nrightarrow [\![\perp]\!]$

$$(u,\star) \mapsto \mathbf{Hom}_{![\![\Gamma]\!]}(\langle\rangle, u)$$

This profunctor corresponds to a generalized version of the one species $\mathbb{1}$

Recall that a (left) group action on a set can be seen as a (covariant) functor $F : \mathbf{G} \to \mathbf{Set}$ where $\mathbf{G}$ is the single object category corresponding to the group in question. By viewing generalized species as groupoid actions (i.e. functors from a groupoid to **Set**), we are able to generalize both the notions of molecular and flat species, they correspond respectively to transitive actions and free (or semi-regular) actions.

Theorem 4.1. *Given a species $\mathbb{F} : !\mathbf{A} \nrightarrow \mathbf{B}$, the following are equivalent:*

1. *$\mathbb{F}$ is molecular i.e. for all species $\mathbb{G}$ and $\mathbb{H}$, $\mathbb{F} = \mathbb{G} + \mathbb{H}$ implies $\mathbb{G} = \mathbb{0}$ or $\mathbb{H} = \mathbb{0}$;*
2. *$\mathbb{F}$ is a transitive groupoid action i.e. for all $(u,b),(u',b')$ and $x \in \mathbb{F}(u,b)$, $x' \in \mathbb{F}(u',b')$, there exists $f : u \to u'$, $g : b' \to b$ such that $\mathbb{F}(f,g)(x) = x'$;*
3. *There exists a unique connected component $\mathscr{C}$ of $!\mathbf{A} \times \mathbf{B}^{op}$ on which $\mathbb{F}$ is non-empty and there exists $(u_0,b_0) \in \mathscr{C}$ and subgroups $H_1 \leq \mathbf{Hom}_{!\mathbf{A}}(u_0,u_0), H_2 \leq \mathbf{Hom}_{\mathbf{B}}(b_0,b_0)$ such that $\mathbb{F}$ is isomorphic to the species*

$$(u,b) \mapsto \mathbf{Hom}_{!\mathbf{A}}(u_0,u) \times \mathbf{Hom}_{\mathbf{B}}(b_0,b)/(H_1 \times H_2).$$

In our case, we can show that we obtain all generalized flat species and we can define integrals in the semantics for all our terms but there are no corresponding terms in the syntax and solutions of differential equations are usually constructed as formal infinite sums of molecular species. We hence first need to add new primitives in the language in order to be able to express all generalized molecular species. To do so, we need a finer understanding of what it means to take the quotient by a subgroup for the variables in a λ-term.

Future Work

One preliminary direction to investigate is which new primitives we can add to our language of terms in order to obtain all possible combinatorial species. Since we are now able to define species of higher order, we hope to gain insight into the existing theory of combinatorial species as well as get new ways of realizing these higher order species by enriching the theory. We also want to have a finer understanding of reduction in this setting, i.e. what 2-cells can we obtain between the interpretation of two equivalent terms?

References

[1] F. Bergeron, G. Labelle & P. Leroux (1998): *Combinatorial species and tree-like structures*. Encyclopedia of Mathematics and its Applications 67, Cambridge University Press, Cambridge. Translated from the 1994 French original by Margaret Readdy, With a foreword by Gian-Carlo Rota.

[2] Jacques Carette & Gordon Uszkay (2008): *Species: making analytic functors practical for functional programming*.

[3] Thomas Ehrhard (2016): *An introduction to Differential Linear Logic: proof-nets, models and antiderivatives*. CoRR abs/1606.01642. Available at http://arxiv.org/abs/1606.01642.

[4] M. Fiore, N. Gambino, M. Hyland & G. Winskel (2008): *The Cartesian closed bicategory of generalised species of structures*. J. Lond. Math. Soc. (2) 77(1), pp. 203–220, doi:10.1112/jlms/jdm096. Available at http://dx.doi.org/10.1112/jlms/jdm096.

[5] Jean-Yves Girard (1988): *Normal functors, power series and λ-calculus*. Ann. Pure Appl. Logic 37(2), pp. 129–177, doi:10.1016/0168-0072(88)90025-5. Available at http://dx.doi.org/10.1016/0168-0072(88)90025-5.

[6] André Joyal (1981): *Une théorie combinatoire des séries formelles*. Adv. in Math. 42(1), pp. 1–82, doi:10.1016/0001-8708(81)90052-9. Available at http://dx.doi.org/10.1016/0001-8708(81)90052-9.

[7] Joachim Kock (2012): *Data Types with Symmetries and Polynomial Functors over Groupoids*. Electronic Notes in Theoretical Computer Science 286, pp. 351 – 365, doi:https://doi.org/10.1016/j.entcs.2013.01.001. Available at http://www.sciencedirect.com/science/article/pii/S1571066113000029. Proceedings of the 28th Conference on the Mathematical Foundations of Programming Semantics (MFPS XXVIII).

[8] Brent A. Yorgey (2010): *Species and Functors and Types, Oh My!* SIGPLAN Not. 45(11), pp. 147–158, doi:10.1145/2088456.1863542. Available at http://doi.acm.org/10.1145/2088456.1863542.