Substructural Type Theories Modelled by Polynomial Functors (Talk Abstract)

The category **Poly**, of polynomial endofunctors on Set [9], is known to have a rich array of categorical structures [10] and has featured extensively in work on applied category theory. The perspective of **Poly**'s morphisms as bidirectional transformations leads to them being a useful model for processes following a request/response pattern, with a covariant component propagating requests and a contravariant part returning responses, whose types are dependent over their correspondent requests [11, 5]. Polynomial functors are closed under composition, acting as a kind of sequencing operation, interleaving requests and responses into more complex interaction patterns. While this can be a powerful semantic domain for modelling interactive systems, notations for defining polynomials and their morphisms can be cumbersome in complex situations, obfuscating the underlying information flow being expressed. This work aims to find a suitable typed language for defining polynomials and their morphisms in an ergonomic fashion.

Von Glehn[12] shows how **Poly** can be used as a model of intuitionistic dependent type theory (ITT). However the structural nature of intuitionistic type theory means that many of the features that make **Poly** interesting as a category of study are inaccessible within this syntactic presentation. In particular the bidirectional nature of morphisms in **Poly** is obfuscated, with much of the logical structure merely reflecting that of the base category (i.e. **Set**). We explore a selection of extensions to dependent type theory with **Poly** as an intended model, which allow for practical axiomatisations of more specialised structures in **Poly**.

Spatial Modalities and Effects. We begin by extending ITT with a pair of adjoint modalities, related to those of spatial type theory [7]:

$$\begin{tikzcd}
\mathbf{Poly} \arrow[rr, "\mathbf{Poly}(\text{よ}^1{,}-)" description] & & \mathbf{Set} \arrow[ll, bend right, "\mathbf{disc}\,:\,X \mapsto X\text{よ}^1"'] \arrow[ll, bend left, "\mathbf{codisc}\,:\,X \mapsto X\text{よ}^0"] \arrow[ll, phantom, shift right=3, "\perp"] \arrow[ll, phantom, shift left=3, "\perp"]
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{Poly} \arrow[rr, bend left, "\flat\,=\,\mathbf{disc}\,\circ\,\mathbf{Poly}(\text{よ}^1{,}-)"] \arrow[rr, phantom, "\perp"] & & \mathbf{Poly} \arrow[ll, bend left, "\sharp\,=\,\mathbf{codisc}\,\circ\,\mathbf{Poly}(\text{よ}^1{,}-)"]
\end{tikzcd}$$

Whereas spatial type theory employs dual contexts to handle substitutions through the comonadic modality $\flat$, we grade variables in the context with mode annotations to allow a more fine-grained assignment of modalities.

These facilitate the definition of "response types" which provide a way for types to refer to the contravariant component of polynomials, and lead to introduction and elimination rules for the composition product of functors.

$$\frac{\Gamma \vdash A \text{ type} \quad \Gamma \vdash e : \sharp A}{\Gamma \vdash \mathsf{Resp}(A, e) \text{ type}} \qquad \frac{\Gamma \vdash p : \sharp A \quad \Gamma,\ x : \mathsf{Resp}(A, p) \vdash e : B}{\Gamma \vdash \mathsf{perform}\ p\ \mathsf{with}\ x \mapsto e : A \lhd B}$$

$$\frac{\Gamma \vdash e : A \lhd B \qquad \Gamma,\ x :^{\sharp} A,\ k : \mathsf{Resp}(A, x) \to B \vdash h : C}{\Gamma \vdash \mathsf{handle}\ e\ \mathsf{with}\ (x, k) \mapsto h : C}$$

This syntax is deliberately chosen to recall the concept of algebraic effects and handlers. Indeed, when we further extend the theory with a free monad type constructor, which can be defined as a fixed-point $E^* \cong E \lhd E^* + I$, we can view terms of the type $E^* \lhd A$ as computations of type A in the presence of effects from a signature described by E, as has been noted in previous work [2, 4]. Generalising this perspective we think of the type $E \lhd A$ as describing a computation of type A in the presence of a 'one-shot' effect of type E. From this point of view we could consider our language as a first-class theory of effects where types and effect signatures are one and the same.

Graded Linear Type Theory. The Dirichlet product (given by the Day convolution of the underlying functors) can be interpreted as modelling a multiplicative linear product between polynomials. Unlike in the previous theory this operation does not fit into ITT with unrestricted cartesian contexts; but where the operation of context extension in the previous models can be seen as generalising the cartesian product of polynomials, the Dirichlet product has a similar generalisation to a dependent linear product, leading us to consider a theory with linear contexts, wherein every variable must be used exactly once.

In this system we find that the flat modality, $\flat$, plays a similar role to the notion of erasure in other linear dependent systems (such as Quantitative Type Theory [1]) mediating a distinction between term-level and type-level usage. Due to the comonadic nature of this modality, and the somewhat surprising departure this implies from the usual dynamics of erasure in other languages, we refer to the modality in this context as 'coerasure'.

This structure, together with a suitable exponential comonad (turning cartesian products into Dirichlet products), forms the basis of a model for a graded linear/non-linear dependent type theory, supporting full type dependency between both linear and non-linear types, and retaining the extensions discussed in the previous section to support response types and the composition product.

$$\begin{tikzcd}
\mathbf{Poly} \arrow[rr, "\sharp"] \arrow[dr, "\flat"'] & \arrow[d, phantom, "\cong"] & \mathbf{Poly} \arrow[dl, "!"] \\
& \mathbf{Poly} &
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{Poly} \times \mathbf{Poly} \arrow[r, "(\times)"] \arrow[d, "!\times!"'] \arrow[dr, phantom, "\cong"] & \mathbf{Poly} \arrow[d, "!"] \\
\mathbf{Poly} \times \mathbf{Poly} \arrow[r, "(\otimes)"'] & \mathbf{Poly}
\end{tikzcd}$$

Bunched Type Theory. Finally we show how **Poly** can be used as a model for a (dependent) labelled bunched theory [8, 3] inspired by the bunched type theory for parameterised spectra of Riley [6].

Similar to the models discussed previously here, Riley's theory makes use of a spatial modality to mediate between type-level dependency and term-level usage. However while the additive setting of Riley's intended model means that $\flat$ and $\sharp$ coincide, we require a more fine-grained approach where $\flat$ provides a notion of coerasure in multiplicative context regions, and $\sharp$ models erasure in additive regions. While the inclusion of an additional modality adds an additional level of complexity, we note that the way in which type dependency is modelled between polynomials is more restricted than that of the models Riley considers, allowing us to simplify other aspects of the theory.

Extending our notion of context to admit both cartesian extension and linear Dirichlet extension provides a synthesis of the expressive power of the cartesian and linear languages we have discussed thus-far, and results in a theory which we conjecture is complete for a category of polynomials defined over a base syntactic model.

References

[1] Robert Atkey. Syntax and semantics of quantitative type theory. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, page 56–65, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355834. doi: 10.1145/3209108.3209189. URL `https://doi.org/10.1145/3209108.3209189`.

[2] Casper Bach Poulsen and Cas van der Rest. Hefty algebras: Modular elaboration of higher-order algebraic effects. *Proc. ACM Program. Lang.*, 7(POPL), January 2023. doi: 10.1145/3571255. URL `https://doi.org/10.1145/3571255`.

[3] Didier Galmiche, Michel Marti, and Daniel Méry. Relating labelled and label-free bunched calculi in bi logic. In Serenella Cerrito and Andrei Popescu, editors, *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 130–146, Cham, 2019. Springer International Publishing. ISBN 978-3-030-29026-9.

[4] Harrison Grodin and David Spivak. Poly-morphic effect handlers (blog post). 2024. URL `https://topos.institute/blog/2024-01-03-algebraic-effect-handlers/`.

[5] Nelson Niu and David I. Spivak. *Polynomial Functors: A Mathematical Theory of Interaction.* London Mathematical Society Lecture Note Series. Cambridge University Press, 2025.

[6] Mitchell Riley. *A Bunched Homotopy Type Theory for Synthetic Stable Homotopy Theory.* PhD thesis, 2022. URL `https://digitalcollections.wesleyan.edu/object/ir-3269`.

[7] Michael Shulman. Brouwer's fixed-point theorem in real-cohesive homotopy type theory. *Mathematical Structures in Computer Science*, 28(6):856–941, 2018. doi: 10.1017/S0960129517000147. URL `https://doi.org/10.1017/S0960129517000147`.

[8] Thomas Somers. Bunchless lambda-calculus for bunched implications. Master's thesis, 2024. URL `https://www.cs.ru.nl/masters-theses/2024/T_Somers___Bunchless_%CE%BB-calculus_for_Bunched_Implications.pdf`.

[9] David I. Spivak. Poly: An abundant categorical setting for mode-dependent dynamics, 2020. URL `https://arxiv.org/abs/2005.01894`.

[10] David I. Spivak. A reference for categorical structures on **Poly**, 2025. URL `https://arxiv.org/abs/2202.00534`.

[11] Andre Videla and Matteo Capucci. Lenses for composable servers, 2022. URL `https://arxiv.org/abs/2203.15633`.

[12] Tamara Von Glehn. *Polynomials and models of type theory.* PhD thesis, Apollo - University of Cambridge Repository, 2015. URL `https://www.repository.cam.ac.uk/handle/1810/254394`.