Modalities and Parametric Adjoints

Daniel Gratzer,[1] Evan Cavallo,[2] G.A. Kavvos,[3] Adrien Guatto,[4] Lars Birkedal[1]

[1] Aarhus University
gratzer@cs.au.dk, birkedal@cs.au.dk
[2] Stockholm University
evan.cavallo@math.su.se
[3] University of Bristol
alex.kavvos@bristol.ac.uk
[4] Université de Paris, CNRS, IRIF
guatto@irif.fr

Recently, a line of modal type theories centering on 'Fitch style' modalities has been proposed [1, 3, 6, 10]. These type theories incorporate a non-fibered modality which behaves like a right adjoint. Specifically, Fitch style type theories pair a modality $\Box$ with a functor on contexts 🔒 to form a *dependent adjunction*, whose transpositions constitute the introduction and elimination rules:

$$\frac{\Gamma.🔒 \vdash M : A}{\Gamma \vdash \mathsf{mod}(M) : \Box A} \qquad \frac{\Gamma \vdash M : \Box A}{\Gamma.🔒 \vdash \mathsf{unmod}(M) : A}$$

Requiring that these operations form a bijection provides β and η rules for $\Box$. Moreover, the introduction rule is evidently stable under substitution; categorically, this is the the naturality of the bijection in Γ. Unfortunately, the same cannot be said for the elimination principle $\mathsf{unmod}(-)$. A type theorist will immediately identify the "non-general" context in the conclusion and worry that it will prove impossible to commute an arbitrary substitution past $\mathsf{unmod}(-)$. To address this, prior Fitch style type theories have adopted slight variations on the rule, each baking in the bare minimum to ensure the admissibility of substitution.

While it provides a convenient syntax, this approach is brittle, with each modification to the modal apparatus requiring a full redesign. Even restricting attention to a single modal type theory, the resultant syntax cannot be used effectively as an internal language: the proof of admissibility of substitution requires induction not just on terms, but on the definable substitutions. When we use the calculus as an internal language, we add in additional substitutions from the model to more effectively capture the particulars of this situation. In so doing, however, we disrupt the substitution property of our type theory: a lemma proved in one context can no longer be freely applied in a different context, resulting in a type theory that is much less useful. While other solutions to this problem have proposed, most notably a weakening of the elimination rule [9], it has remained unknown how to combine even two common Fitch style modalities such as $\Box$ and $\triangleright$ [7] in one dependent type theory.

We address this state of affairs by assuming additional structure, that of a parametric adjunction, which reconciles the strong Fitch style elimination rule with substitution. We thereby contribute FitchTT, a modal type theory which can support an arbitrary collection of Fitch style modalities and natural transformations between them [8]. It is a small step from one parametric adjoint modality to full FitchTT, but this is testament to the utility of parametric adjoints in structuring the theory. Indeed, FitchTT is capable of containing multiple interacting modalities such as the aforementioned $\Box$ and $\triangleright$ without the difficulties of prior approaches.

More than this, the extra structure of parametric adjoints is latent in all prior Fitch calculi, and their presence in the initial models of these type theories accounts for the admissibility of substitution. As a result, FitchTT conservatively extends DRA [3] and embeds in $\mathsf{MLTT}_{🔒}$ [10].

Furthermore, this extra structure allows us to systematically rederive the syntax of a single-clock variant of Clocked Type Theory [1] and parametric type theory [4] in a uniform setting.

The special case of functions To motivate the role of parametric adjoints in Fitch style modalities, we focus on a concrete modality: exponentiation by a closed type $\mathfrak{C}$. Specializing the above rules with $\Box A = \mathfrak{C} \to A$ and $\Gamma.\text{🔒} = \Gamma.\mathfrak{C}$, we see that the introduction rule is the familiar introduction rule of dependent products, but the elimination rule is more surprising:

$$\frac{\Gamma \vdash M : \mathfrak{C} \to A}{\Gamma.\mathfrak{C} \vdash \mathsf{unmod}(M) : A}$$

This rule is equivalent to the application rule because $[\Delta, \Gamma.\mathfrak{C}] \cong [\Delta, \Gamma] \times [\Delta, \mathbf{1}.\mathfrak{C}]$. We first bundle a substitution $r \colon \Delta \longrightarrow \mathbf{1}.\mathfrak{C}$ with Δ and view the pairing as an object in the slice category $\mathbf{Cx}/\mathfrak{C}$. By taking $\Gamma.\mathfrak{C}$ as another object over $\mathbf{1}.\mathfrak{C}$ by projection, we can rewrite this isomorphism in a more compact form:

$$[\Delta, \Gamma]_{\mathbf{Cx}} \cong [(\Delta, r), (\Gamma.\mathfrak{C}, \mathbf{v}_k)]_{\mathbf{Cx}/\mathfrak{C}}$$

Written this way, we see that $-.\mathfrak{C}$ is a right adjoint, not as a functor $\mathbf{Cx} \longrightarrow \mathbf{Cx}$ but as a functor $\mathbf{Cx} \longrightarrow \mathbf{Cx}/\mathfrak{C}$. More concisely, $-.\mathfrak{C}$ is a parametric right adjoint (PRA):

Definition 1. $F \colon \mathcal{C} \longrightarrow \mathcal{D}$ is a parametric right adjoint if $F/\mathbf{1} \colon \mathcal{C} \longrightarrow \mathcal{D}/F(\mathbf{1})$ is a right adjoint.

In the case of $-.\mathfrak{C}$, the left adjoint U is forgetful functor $\mathbf{Cx}/\mathfrak{C} \longrightarrow \mathbf{Cx}$ which sends (Γ, r) to Γ. We now restate the traditional application rule purely in terms of this parametric adjunction:

$$\frac{r \colon \Gamma \longrightarrow \mathbf{1}.\mathfrak{C} \qquad U(\Gamma, r) \vdash M : \mathfrak{C} \to A}{\Gamma \vdash M\langle r\rangle : A[\eta[r]]}$$

Unlike the rule for $\mathsf{unmod}(-)$ specialized to $\mathfrak{C} \to -$, this rule is stable under substitution. Recalling that $U(\Gamma, r) = \Gamma$, this rule becomes precisely the familiar application rule.

Generalizing with PRAs Taking our cue from this special example, we consider a general Fitch style modality $\langle \mu \mid - \rangle$ whose left adjoint on contexts $-.\{\mu\}$ is a parametric right adjoint. We adopt the notation $\Gamma/(r : \mu) = U(\Gamma, r)$ for the parametric left adjoint to $-.\{\mu\}$ by analogy with the construct used in nominal and parametric type theories [2, 4, 5].

The introduction rule for $\langle \mu \mid - \rangle$ remains unchanged, but we now take the modified variant of the application rule for our elimination rule:

$$\frac{r \colon \Gamma \longrightarrow \mathbf{1}.\{\mu\} \qquad \Gamma/(r : \mu) \vdash M : \langle \mu \mid A \rangle}{\Gamma \vdash M \mathbin{@} r : A[\eta[r]]}$$

We may equip this rule with β and η rules which closely mirror those of dependent products. We further observe that $M \mathbin{@} r$ is interderivable with $\mathsf{unmod}(M)$, but stable under substitution.

Unlike the ad hoc variants of $\mathsf{unmod}(-)$ used in prior Fitch style type theories, this rule scales to multiple modalities. In fact, no issues arise if we allow any strict 2-category of modes, modalities, and natural transformations [11] between them, provided that we require that each modality is equipped with a left adjoint on contexts which is itself a PRA.

References

[1] Patrick Bahr, Hans Bugge Grathwohl, and Rasmus Ejlers Møgelberg. 2017. The clocks are ticking: No more delays!. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. IEEE. https://doi.org/10.1109/LICS.2017.8005097

[2] Jean-Philippe Bernardy, Thierry Coquand, and Guilhem Moulin. 2015. A Presheaf Model of Parametric Type Theory. *Electronic Notes in Theoretical Computer Science* 319 (2015), 67–82. https://doi.org/10.1016/j.entcs.2015.12.006

[3] Lars Birkedal, Ranald Clouston, Bassel Mannaa, Rasmus Ejlers Møgelberg, Andrew M. Pitts, and Bas Spitters. 2020. Modal dependent type theory and dependent right adjoints. *Mathematical Structures in Computer Science* 30, 2 (2020), 118–138. https://doi.org/10.1017/S0960129519000197 arXiv:1804.05236

[4] Evan Cavallo and Robert Harper. 2020. Internal Parametricity for Cubical Type Theory. In *28th EACSL Annual Conference on Computer Science Logic (CSL 2020) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 152)*, Maribel Fernández and Anca Muscholl (Eds.). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 13:1–13:17. https://doi.org/10.4230/LIPIcs.CSL.2020.13

[5] James Cheney. 2012. A dependent nominal type theory. *Log. Methods Comput. Sci.* 8, 1 (2012). https://doi.org/10.2168/LMCS-8(1:8)2012

[6] Ranald Clouston. 2018. Fitch-Style Modal Lambda Calculi. In *Foundations of Software Science and Computation Structures*, Christel Baier and Ugo Dal Lago (Eds.). Springer International Publishing, 258–275.

[7] Ranald Clouston, Aleš Bizjak, Hans Bugge Grathwohl, and Lars Birkedal. 2015. Programming and Reasoning with Guarded Recursion for Coinductive Types. In *Foundations of Software Science and Computation Structures* (Berlin, Heidelberg), Andrew Pitts (Ed.). Springer Berlin Heidelberg, 407–421.

[8] Daniel Gratzer, Evan Cavallo, G.A. Kavvos, Adrien Guatto, and Lars Birkedal. 2021. Modalities and Parametric Adjoints. https://jozefg.github.io/papers/modalities-and-parametric-adjoints.pdf Under review.

[9] Daniel Gratzer, G.A. Kavvos, Andreas Nuyts, and Lars Birkedal. 2020. Multimodal Dependent Type Theory. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '20)*. ACM. https://doi.org/10.1145/3373718.3394736

[10] Daniel Gratzer, Jonathan Sterling, and Lars Birkedal. 2019. Implementing a Modal Dependent Type Theory. *Proc. ACM Program. Lang.* 3 (2019). Issue ICFP. https://doi.org/10.1145/3341711

[11] Daniel R. Licata and Michael Shulman. 2016. Adjoint Logic with a 2-Category of Modes. In *Logical Foundations of Computer Science*, Sergei Artemov and Anil Nerode (Eds.). Springer International Publishing, 219–235. https://doi.org/10.1007/978-3-319-27683-0_16