Proof-relevant Kripke semantics of $\mu\mathsf{ML}$ via fixpoints of containers

Ezra Schoen[1] and Sean Watters[1,2]

[1] *University of Strathclyde, UK*
[2] *Coherence Research Ltd.*

Introduction

The modal μ-calculus $\mu\mathsf{ML}$ [Koz83] is the logic obtained by adding least and greatest fixed point quantifiers to modal logic. It subsumes many temporal and dynamic logics such as $\mathsf{PDL}, \mathsf{LTL}, \mathsf{CTL}$ and CTL^*. It is usually presented classically, but two constructive variants have recently been proposed ([Pac23] and [AG25]).

In this talk, we present an ongoing project on formalising a constructive variant of the modal μ-calculus inside Agda [Nor09], a proof assistant based on dependent type theory. We use a *proof-relevant Kripke semantics*: the semantics of the $\Diamond$ modality does not require the mere *existence* of a successor, but a *choice* of successor.

This naturally gives rise to a type-valued semantics of modal formulas, however handling the fixpoint quantifiers μ, ν requires some subtlety. A naive attempt at a shallow embedding using Agda's built-in induction and coinduction fails, because the least and greatest fixpoints are not well-defined for arbitrary (indexed) endofunctors. Instead, we use the machinery of *indexed containers* [Alt+15] to handle the fixed points. That is, we introduce a semantics $[\![-]\!]_c$ which maps a fixpoint formula to an indexed container; indexed containers represent a subclass of type families that have the required structure to interpret the propositional connectives $\wedge$ and $\vee$, modal operators, *and* the fixpoint quantifiers. We currently do not include implication or negation in our logic, however we have good reason to believe that our approach can support these also, despite the commonly held belief to the contrary in the presence of fixpoints — we will discuss this towards the end.

The result is a version of Kripke semantics that is:

Proof-Relevant: The semantics $[\![\varphi]\!]$ of a formula maps a model $\mathcal{M}$ with worlds W and a world $w : W$ to a *type* of witnesses of $\mathcal{M}, w \Vdash \varphi$; such witnesses may be thought of as winning strategies for Verifier in a semantics game.

Constructive: Because everything is formulated inside type theory, we work entirely in a constructive setting. As a result, constructivity (e.g. failure of LEM, non-duality of $\Box$ and $\Diamond$) is inherited straightforwardly from the metatheory.

A similar approach was recently taken by Todorov and Bach [TP24], who applied $\mu\mathsf{ML}$ for verification of properties of effectful programs, modelled as (co)recursively defined trees in Agda. While the technique for ensuring the fixpoints are well-defined is the same between them and us — we both employ indexed containers — their models (and therefore their notion of modality) and ultimate goals were different from ours. While they do handle implication and negation, they do so using non-constructive rewrite rules (e.g. material implication).

Indexed Containers for Kripke Semantics

The theory of containers [AAG03; Abb03] is a formalism for representing polynomial endofunctors. They are well-known amongst the type theory community for their ability to capture strictly-positive inductive data types [AAG05]. The notion of containers can be extended to that of *indexed* containers, which in turn represent indexed polynomial functors and strictly-positive type *families* [Alt+15].

Proposition ([Alt+15])**.** *For all I, J : Type, there is a type $\mathsf{ICont}(I, J)$ of "(I, J)-indexed containers", such that:*

1. *The set of (I, J)-indexed containers forms a category which has initial and terminal objects, and is closed under products and co-products.* □

2. *There are operations* $\mu, \nu : \mathsf{ICont}(I + J, J) \to \mathsf{ICont}(I, J)$ *which compute initial algebras (least fixpoints) and terminal co-algebras (greatest fixpoints).* □

3. *There exists a full and faithful functor* $[\![-]\!]_F : \mathsf{ICont}(I, J) \to ((I \to \mathsf{Type}) \to (J \to \mathsf{Type}))$ *that interprets an indexed container as an indexed endofunctor on* Type*, and which preserves the above structure.* □

The constructive proofs of the above closure properties are exactly the *combinators* which construct the product container, least fixpoint container, etc. These combinators provide a natural syntax (moreover, a fixpoint calculus) for constructing containers; this is a ubiquitous approach in containers literature which dates at least as far back as [AAG04].

Our use of *indexed* containers is crucial for handling the Kripke structure. The meaning of a formula φ depends on a choice of world $w : W$, therefore we require that $[\![\varphi]\!] : W \to \mathsf{Type}$ (when φ is a sentence). To handle formulas with n free variables, we use a "semantic context" $(W \to \mathsf{Type})^n$ of interpretations for the free variables (implemented as a length-indexed list). Thus we have that the meaning of a formula φ with n free variables should have type $[\![\varphi]\!] : (W \to \mathsf{Type})^n \to (W \to \mathsf{Type})$. By uncurrying, we can rewrite this type as $((\mathsf{Fin}(n) \times W) \to \mathsf{Type}) \to (W \to \mathsf{Type})$, where $\mathsf{Fin}(n)$ is a finite set of size n. This is now clearly an indexed functor, which we will represent by an $\mathsf{ICont}(\mathsf{Fin}(n) \times W, W)$.

Theorem. *Fix a Kripke model* (W, R, V) *where* $W : \mathsf{Type}$*,* $R : W^2 \to \mathsf{Type}$*, and* $V : A \times W \to \mathsf{Type}$ *for some set of atomic propositions* $A : \mathsf{Type}$*. Then for any formula* φ *with at most* n *free variables, there is an indexed container* $[\![\varphi]\!]_c : \mathsf{ICont}(\mathsf{Fin}(n) \times W, W)$ *which denotes its semantics (via composition with* $[\![-]\!]_F$*, yielding* $[\![-]\!] = [\![-]\!]_F \circ [\![-]\!]_c$*).* □

Intuitively, the type $[\![\varphi]\!](w)$ corresponds to the type of "witnesses of $w \Vdash \varphi$". The full definition of $[\![-]\!]_c$ directly maps formula symbols to the associated combinators of containers (e.g. $\wedge$ to products, μ to the initial algebra). For the modal operators, we construct operations on containers such that ultimately, a witness of $w \Vdash \Diamond\varphi$ is a pair of a successor w' together with a witness of $w' \Vdash \varphi$. For $\Box$, we obtain that a witness of $w \Vdash \Box\varphi$ is a function mapping each successor w' of w to a witness of $w' \Vdash \varphi$.

Open Questions

Proof Theory. What we have so far is a semantics for (a fragment of) the modal μ-calculus; what is lacking is a proof system. We also do not know if this semantics is correct or complete for (the strictly positive fragments of) either the constructive μ-calculus [Pac23] or iL_μ from [AG25], or represents a third candidate for an intuitionistic μ-calculus.

Interestingly, there seems to be a tight relationship between the container semantics and (non-wellfounded) proofs. A promising example of is given by the contrast between the validities $\top$ and $\mu X.\top \vee X$. As types, $[\![\top]\!]$ is given by the unit type $\mathbf{1}$, which has a single point $*$. On the other hand, $[\![\mu X.\top \vee X]\!]$ is given by the least fixed point of $X \mapsto \mathbf{1} + X$, which is the set $\mathbb{N}$ of natural numbers. Correspondingly, in (non-wellfounded) proof systems such as $\mathsf{T}^{\mathsf{pre}}_\mu$ from [Stu08], the sequent $\top$ has a unique proof; whereas the sequent $\mu X.\top \vee X$ has countably many proofs, corresponding to the number of unfoldings of the fixed point variable before the left disjoint $\top$ is chosen.

Implication and Negation. Morphisms $C \Rightarrow D$ in the category of containers correspond to natural transformations of polynomial functors, or *polymorphic* functions between strictly positive data types. From the fact that $[\![-]\!]_F$ is full and faithful, we know that all such polymorphic functions between the represented data types are captured this way. Container morphisms are a natural candidate for modelling *entailment*: a judgement $\phi_0, \ldots \phi_n \vdash \psi$ would be represented by a morphism $([\![\phi_0]\!]_c \times \ldots \times [\![\phi_n]\!]_c) \Rightarrow [\![\psi]\!]_c$. This suggests a pathway to reasoning about entailment without going beyond strictly positive datatypes (and formulas).

Rather unintuitively, the category of (non-indexed) containers is also Cartesian closed [ALS10]. That is, for containers C, D, there is a container D^C which internalises the set of morphisms $C \Rightarrow D$ in the sense that there is an adjunction $(- \times X) \dashv (-^X)$. For this adjunction, the forwards and backwards maps of the natural isomorphism of homsets correspond to currying and uncurrying, respectively. This means that, at least in some sense, polymorphic functions can themselves be viewed as strictly positive data types.

An obvious further question is whether these exponential containers can likewise model constructive implication — we believe so, at least in some sense. In her PhD thesis [Von15, Ch. 4], von Glehn showed that Altenkirch, Levy, and Staton's construction of the exponentials of non-indexed containers can be

extended to even give a model of (intensional) *dependent* function types (AKA Π-types), despite the category not being locally Cartesian closed. The key result of von Glehn's thesis is that containers form a model of Martin-Löf's dependent type theory. This goes beyond our requirement for simple function types/implication; the presence of Π- and Σ-types tell us that containers can even model the first-order universal and existential quantifiers.

The question for us at the moment is whether there is a suitable notion of exponentials for *indexed* containers, at minimum for our specific case of $\mathsf{ICont}(\mathsf{Fin}(n) \times W, W,)$. Our current belief is that this may only exist for certain subcategories of indexed containers; i.e. not all morphisms will be exponentiable. We are hopeful to still obtain an interesting notion of implication in the end, but this remains a work-in-progress. Even before moving to the indexed setting, it would be helpful to more concretely understand the interaction between fixpoints and exponentials in the category of non-indexed containers, since their mutual presence is rather surprising.

Assuming we are able to achieve a sensible model of implication, negation would be modelled via implication of falsity, as is standard for intuitionistic logic. We do expect this to be a non-classical model; the induced double negation map should not be an isomorphism.

Modal Type Theory. Finally, under the Curry-Howard correspondence, any proof system for a modal μ-calculus corresponds to some kind of modal type theory (see e.g. [Gra23; GSB19]) — logics are the proof-irrelevant fragments of type theories. Moreover, there is a class of one-world Kripke models which reduces the semantics of modal operators to no-op. Thus, we hope that:

Conjecture. *There is a modal type theory whose proof-irrelevant fragment is an intuitionistic variant of the modal μ-calculus, and whose non-modal fragment is a suitable type theory, such that their mutual proof-irrelevant, non-modal fragment is intuitionistic propositional logic.*

We have three main goals going forward: to continue to investigate the status of implication and negation in our semantics, to construct a proof system for reasoning about validity and/or entailment, and ultimately to expand the proof system into a suitable modal type theory for our semantics. We anticipate such a modal type theory having applications for formal verification of coloured Petri nets [Jen87; GS20], and generalisations thereof. A yet-further extension, inspired by von Glehn's work, would be to consider a constructive first-order modal μ-calculus, with the quantifiers being modelled by Π- and Σ-types in the category of containers.

References

[Abb03] Michael Gordon Abbott. 'Categories of containers'. PhD thesis. University of Leicester, England, UK, 2003. URL: https://figshare.le.ac.uk/articles/thesis/Categories_of_containers/10155104/files/18301229.pdf.

[AAG03] Michael Gordon Abbott, Thorsten Altenkirch and Neil Ghani. 'Categories of Containers'. In: *Foundations of Software Science and Computational Structures, 6th International Conference, FOSSACS 2003 Held as Part of the Joint European Conference on Theory and Practice of Software, ETAPS 2003, Warsaw, Poland, April 7-11, 2003, Proceedings*. Ed. by Andrew D. Gordon. Vol. 2620. Lecture Notes in Computer Science. Springer, 2003, pp. 23–38. DOI: 10.1007/3-540-36576-1_2.

[AAG04] Michael Gordon Abbott, Thorsten Altenkirch and Neil Ghani. 'Representing Nested Inductive Types Using W-Types'. In: *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004. Proceedings*. Ed. by Josep Díaz, Juhani Karhumäki, Arto Lepistö and Donald Sannella. Lecture Notes in Computer Science. Springer, 2004, pp. 59–71. DOI: 10.1007/978-3-540-27836-8_8.

[AAG05] Michael Gordon Abbott, Thorsten Altenkirch and Neil Ghani. 'Containers: Constructing strictly positive types'. In: *Theor. Comput. Sci.* 342.1 (2005), pp. 3–27. DOI: 10.1016/J.TCS.2005.06.002.

[AG25] Bahareh Afshari and Lide Grotenhuis. 'Intuitionistic μ-Calculus with the Lewis Arrow'. In: *Automated Reasoning with Analytic Tableaux and Related Methods - 34th International Conference, TABLEAUX 2025, Reykjavik, Iceland, September 27-29, 2025, Proceedings*. Ed. by Gian Luca Pozzato and Tarmo Uustalu. Lecture Notes in Computer Science. Springer, 2025, pp. 374–392. DOI: 10.1007/978-3-032-06085-3_20.

[Alt+15] Thorsten Altenkirch, Neil Ghani, Peter G. Hancock, Conor McBride and Peter Morris. 'Indexed containers'. In: *J. Funct. Program.* 25 (2015). DOI: 10.1017/S095679681500009X.

[ALS10] Thorsten Altenkirch, Paul Blain 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*. Ed. by Fernando Ferreira, Benedikt Löwe, Elvira Mayordomo and Luís Mendes Gomes. Lecture Notes in Computer Science. Springer, 2010, pp. 11–20. DOI: 10.1007/978-3-642-13962-8_2.

[GS20] Fabrizio Genovese and David I. Spivak. 'A Categorical Semantics for Guarded Petri Nets'. In: *Graph Transformation*. Springer International Publishing, 2020, pp. 57–74. ISBN: 9783030513726. DOI: 10.1007/978-3-030-51372-6_4.

[Gra23] Daniel Gratzer. 'Syntax and semantics of modal type theory'. PhD thesis. Aarhus University, Denmark, 2023. URL: https://pure.au.dk/ws/files/428926355/main.pdf.

[GSB19] Daniel Gratzer, Jonathan Sterling and Lars Birkedal. 'Implementing a modal dependent type theory'. In: *Proc. ACM Program. Lang.* 3.ICFP (2019), 107:1–107:29. DOI: 10.1145/3341711.

[Jen87] Kurt Jensen. 'Coloured Petri Nets'. In: *Petri Nets: Central Models and Their Properties*. Ed. by W. Brauer, W. Reisig and G. Rozenberg. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987, pp. 248–299. ISBN: 978-3-540-47919-2. DOI: https://doi.org/10.1007/978-3-540-47919-2_10.

[Koz83] Dexter Kozen. 'Results on the Propositional mu-Calculus'. In: *Theor. Comput. Sci.* 27 (1983), pp. 333–354. DOI: 10.1016/0304-3975(82)90125-6.

[Nor09] Ulf Norell. 'Dependently Typed Programming in Agda'. In: *Advanced Functional Programming: 6th International School, AFP 2008, Heijen, The Netherlands, May 2008, Revised Lectures*. Ed. by Pieter Koopman, Rinus Plasmeijer and Doaitse Swierstra. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 230–266. ISBN: 978-3-642-04652-0. DOI: 10.1007/978-3-642-04652-0_5.

[Pac23] Leonardo Pacheco. 'Game semantics for the constructive μ-calculus'. In: *CoRR* abs/2308.16697 (2023). DOI: 10.48550/ARXIV.2308.16697.

[Stu08] Thomas Studer. 'On the Proof Theory of the Modal mu-Calculus'. In: *Stud Logica* 89.3 (2008), pp. 343–363. DOI: 10.1007/S11225-008-9133-6.

[TP24] Ivan Todorov and Casper Bach Poulsen. 'Modal μ-Calculus for Free in Agda'. In: *Proceedings of the 9th ACM SIGPLAN International Workshop on Type-Driven Development, TyDe 2024, Milan, Italy, 6 September 2024*. Ed. by Sandra Alves and Jesper Cockx. ACM, 2024, pp. 16–28. DOI: 10.1145/3678000.3678202.

[Von15] Tamara Von Glehn. 'Polynomials and models of type theory'. PhD thesis. University of Cambridge, 2015. DOI: 10.17863/CAM.16245. URL: https://www.repository.cam.ac.uk/handle/1810/254394.