Two Guarded Recursive Powerdomains for Applicative Simulation

Rasmus Ejlers Møgelberg and Andrea Vezzosi

IT University of Copenhagen, Denmark
(mogel,avez@itu.dk)

Last year at Types we presented Clocked Cubical Type Theory (CCTT) [5], a type theory combining multi-clocked guarded recursion with Cubical Type Theory. One use case for this type theory is for programming and reasoning with coinductive types, encoding these via guarded recursion. CCTT allows one to do this also for coinductive types defined using higher inductive types, and one can moreover prove that path type equality for these coincides with bisimilarity. Another use is as a meta-language for both operational and denotational models of programming languages. This talk presents a worked example of using both these ideas, and is based on our newly published paper [9].

Guarded Powerdomains in Clocked Cubical Type Theory

Clocked Cubical Type Theory extends Cubical Type Theory with a pre-type of clocks, and for each clock κ, a modality $\triangleright^\kappa$ and a fixed point combinator $\mathsf{fix}^\kappa : (\triangleright^\kappa A \to A) \to A$. One use of the fixed point combinator is to construct guarded recursive types, such as $\mathsf{L}^\kappa A$ satisfying $\mathsf{L}^\kappa A \simeq A + \triangleright^\kappa \mathsf{L}^\kappa A$ as fixed points of maps on the universe. Defining a κ-delay algebra to be a type B with an operations $\triangleright^\kappa B \to B$, $\mathsf{L}^\kappa A$ is the free κ-delay algebra on A. Using quantification over clocks, one can use these to encode coinductive types such as $\mathsf{L}A \stackrel{\mathsf{def}}{=} \forall\kappa.\mathsf{L}^\kappa A$ which is the coinductive solution to $\mathsf{L}A \simeq A + \mathsf{L}A$. This latter is the coinductive delay monad, which can be used to model recursion in type theory, however, working with the guarded recursive variant L^κ gives access to a powerful fixed point operator, and, moreover, guarded recursive types can also have negative occurrences. This has been used for a form of guarded synthetic domain theory, producing models of FPC and PCF and proving these adequate in type theory [8, 11].

This work studies the extension of such models with finite non-determinism. We construct two guarded recursive powerdomains by combining L^κ with the finite powerset functor $\mathsf{P_f}$, defined as a HIT [3], generated by singleton, union and axioms making it the free join-semilatice. The powerdomains are defined as follows

$$\mathsf{P}^\kappa_\Diamond(A) \simeq \mathsf{P_f}(A + \triangleright^\kappa \mathsf{P}^\kappa_\Diamond(A)) \qquad\qquad \mathsf{P}^\kappa_\Box(A) \stackrel{\mathsf{def}}{=} \mathsf{L}^\kappa(\mathsf{P_f}(A))$$

The first of these is a monad defined as a guarded recursive type. An element of this type is a finite set of values of type A and computations that can be run for at least one more step. The subscript refers to may-convergence and is intuitively justified by the fact that it reveals return values for terminated branches even when other branches have not yet terminated. The second is simply the composition of two monads. Unlike $\mathsf{P}^\kappa_\Diamond$, elements of $\mathsf{P}^\kappa_\Box$ do not reveal partial results, but just returns a set of values once all branches have terminated. Unfortunately, $\mathsf{P}^\kappa_\Box$ is not a monad, since the associativity axiom breaks up to step counting. It is, however, a monad up to a notion of weak bisimilarity.

Semantics for the untyped lambda calculus

Both these constructions carry a semilattice structure. In the case of $\mathsf{P}^\kappa_\Diamond$ the union operation is defined using the one for P_f. In the case of $\mathsf{P}^\kappa_\Box$, the union operations evaluates the two given computations in parallel and returns the union once they have both terminated. This means that the delay is the maximum of the delays of the two input computations. Algebraically, $\mathsf{P}^\kappa_\Diamond(A)$ is the free join semilattice and κ-delay algebra on A with no equations between the two structures. For $\mathsf{P}^\kappa_\Box(A)$, the delay algebra structure distributes over the semilattice one, but also satisfies additional non-algebraic interaction equations.

Using the semilattice structures, one can define denotational semantics for the untyped lambda calculus extended with finite non-determinism in the form of an operation $M \,\mathsf{or}\, N$. In both cases, the domain of the denotational semantics is a solution to a guarded recursive domain equation defined as

$$\mathsf{SVal}^\kappa \stackrel{\mathsf{def}}{=} \triangleright^\kappa(\mathsf{SVal}^\kappa \to T(\mathsf{SVal}^\kappa)) \qquad\qquad D^\kappa \stackrel{\mathsf{def}}{=} T(\mathsf{SVal}^\kappa)$$

where T can be instantiated to $\mathsf{P}^\kappa_\Diamond$ and $\mathsf{P}^\kappa_\Box$ (or indeed any monad-like construction with a semilattice structure). This semantics can be proved sound with respect to the standard big-step may- and must operational semantics which we write as $\Downarrow_\Diamond$ and $\Downarrow_\Box$ respectively.

Applicative similarity

As an example application of these powerdomains, we look at how to prove applicative similarity a congruence for the untyped lambda calculus with finite non-determinism. This is usually proved using operational reasoning and Howe's method [7, 6, 4, 2], or in some cases advanced domain theoretic techniques such as Stone duality [10, 1]. Here we build on a proof by Pitts [12], which uses a denotational semantics in domain theory and a relation between syntax and semantics. Our contribution is to extend from the case of pure lambda calculus to finite non-determinism and adapt to guarded synthetic domain theory.

In a few more details, a relation R is an applicative may-simulation if $M\,R\,N$ and $M \Downarrow_\Diamond \lambda x.M'$ implies

$$\exists N'.\, N \Downarrow_\Diamond \lambda y.N' \wedge (\forall(V : \mathsf{Val}).M'[V/x]\,R\,N'[V/x])$$

May-similarity $\leq_\Diamond$ is the greatest may-simulation, and this can be defined in Clocked Cubical Type Theory using a combination of guarded recursion and quantification over clocks, similarly to the coinductive delay monad L. The aim is to show that this is a congruence. Our proof uses a relation $\preceq^\kappa$: $D^\kappa \times \Lambda \to \mathsf{Prop}$ between the denotational semantics mentioned in the previous section, and syntax. The key lemmas state that $[\![M]\!]^\kappa \preceq^\kappa M$ for all closed M, and that $M \leq_\Diamond N$ is equivalent to $\forall\kappa.[\![M]\!]^\kappa \preceq^\kappa N$. We prove a similar result for the case of must-similarity.

Implementation

The results mentioned above have been proved on paper, and only few lemmas have been formally verified in a proof assistant. In the time since this work was completed, Vezzosi has implemented an experimental extension of Agda[1] based on CCTT. Using this it should now be possible to implement these proofs in Agda without much overhead from the paper versions developed in this work. One point of the talk is therefore to announce the Agda branch for CCTT to the Types community.

[1] https://github.com/agda/guarded/tree/forcing-ticks

References

[1] Samson Abramsky. The lazy lambda calculus, research topics in functional programming. 1990.

[2] Ugo Dal Lago, Francesco Gavazzo, and Paul Blain Levy. Effectful applicative bisimilarity: Monads, relators, and howe's method. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2017.

[3] Dan Frumin, Herman Geuvers, Léon Gondelman, and Niels van der Weide. Finite sets in homotopy type theory. In June Andronick and Amy P. Felty, editors, *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2018, Los Angeles, CA, USA, January 8-9, 2018*, pages 201–214. ACM, 2018.

[4] Douglas J Howe. Equality in lazy computation systems. In *LICS*, volume 89, pages 198–203, 1989.

[5] Magnus Baunsgaard Kristensen, Rasmus Ejlers Møgelberg, and Andrea Vezzosi. Greatest hits: Higher inductive types in coinductive definitions via induction under clocks, 2021. arXiv:2102.01969.

[6] Søren B Lassen and Corin S Pitcher. Similarity and bisimilarity for countable non-determinism and higher-order functions. *Electronic Notes in Theoretical Computer Science*, 10:246–266, 1998.

[7] Søren Bøgh Lassen. *Relational reasoning about functions and nondeterminism.* PhD thesis, University of Aarhus, 1998.

[8] Rasmus Ejlers Møgelberg and Marco Paviotti. Denotational semantics of recursive types in synthetic guarded domain theory. In *LICS*, 2016.

[9] Rasmus Ejlers Møgelberg and Andrea Vezzosi. Two guarded recursive powerdomains for applicative simulation. In Ana Sokolova, editor, Proceedings 37th Conference on *Mathematical Foundations of Programming Semantics,* Hybrid: Salzburg, Austria and Online, 30th August - 2nd September, 2021, volume 351 of *Electronic Proceedings in Theoretical Computer Science*, pages 200–217. Open Publishing Association, 2021.

[10] C-HL Ong. Non-determinism in a functional setting. In *[1993] Proceedings Eighth Annual IEEE Symposium on Logic in Computer Science*, pages 275–286. IEEE, 1993.

[11] Marco Paviotti, Rasmus Ejlers Møgelberg, and Lars Birkedal. A model of pcf in guarded type theory. *Electronic Notes in Theoretical Computer Science*, 319:333–349, 2015.

[12] Andrew M Pitts. A note on logical relations between semantics and syntax. *Logic Journal of the IGPL*, 5(4):589–601, 1997.