Lecture 26: Review and Conclusions

Lecturer: Max S. New
Scribe: Matthew Nomura

April 17, 2023

What we've learned

We've learned about categories, functors, natural transformations, etc., but the goal of this class was to learn how to apply these things to computer science. Here are some of the key points I want you to take away:

(I) Models of logics are (structured) posets, and models of programming calculi are (structured) categories. Order theory and category theory are the exact mathematical fields that let us reason about logics and programming calculi, respectively.

(II) In logic, the well-behaved, or "nice", connectives are some kind of meet or join. In type theory/programming calculi, the "nice" type connectives are modeled most generally by representable predicators (covariant or contravariant).

(III) Logics and calculi are weakly inital models. This means that, given the syntax, we can construct a Lindenbaum algebra where the following holds: for any other model M, we can define a semantics $[\![.]\!] : \mathcal{L} \to M$, where $[\![.]\!]$ is unique up to unique isomorphism: there is a unique natural isomorphism to any other homomorphism $\mathcal{L} \to M$.

$$\begin{tikzcd} \mathcal{L} \arrow[r, bend left, "{[\![\cdot]\!]}"] \arrow[r, bend right, dashed] \arrow[r, phantom, "\cong"] & M \end{tikzcd}$$

For logics, we get a simpler characterization: the model is simply unique.

(IV) We can use initiality to prove properties, or metatheorems, about the logics. For example, we've proved:

- Soundness and consistency (like you can't prove falsity starting from an empty context in IPL)
- Convervativity properties. In logic, one such property was that adding the Heyting implication to IPL doesn't allow us to prove any new theorems.
- The law of the excluded middle is not provable in IPL (using soundness)
- Soundness of STT as a logic: $\neg(\cdot \vdash M : 0)$
- Soundness of the equational theory of STT: $\neg(\text{true} \; = \; \text{false})$, by soundness using our set model
- Canonicity: for all $\cdot \vdash M : 1 + 1$, there exists a unique bit $b \in \{0, 1\}$ such that $M = i_b()$. We showed this using the gluing model, which is one of the most powerful techniques we've covered.
- Conservativity of equational theory

(V) These techniques are not just things we can use to study systems after they have been defined. We can use them to go from an idea of a model to a well-behaved calculus where we can prove the above nice results using similar methods. This was an introductory class, so we just got a taste of this, but categorical logic is an ongoing research program that aids in the design of new calculi and logics.

Topics to Check out

These are all topics I considered covering this semester, and so I have provided citations to papers that I recommend as good introductions or references on their categorical semantics, rather than the original references:

(I) Recursion: recursive functions and types [Abramsky and Jung, 1995, Freyd, 1991]

(II) Monads [Moggi, 1991, Wadler, 1993]

(III) Linear logic, which is closely related to call-by-push-value[Benton, 1995, Melliès, 2009]

(IV) First-order logic, higher order logics, modal logics [Shulman, 2016]

(V) Program logic: Hoare logic, Bunched implications, Separation logic [Bizjak and Birkedal, 2018, Pym et al., 2004]

(VI) Dependent type theory (which works as a foundation for proof assistants), Homotopy type theory, Modal type theory[AWODEY, 2018, Gratzer et al., 2022, Seely, 1984]

(VII) Parametric Polymorphism/Parametricity [Hermida et al., 2014]

(VIII) Gradual Typing/Graduality [New and Licata, 2018, New et al., 2019]

References

Samson Abramsky and Achim Jung. Domain theory. *Handbook of logic in computer science*, 10 1995.

STEVE AWODEY. Natural models of homotopy type theory. *Mathematical Structures in Computer Science*, 28(2):241–286, 2018. doi: 10.1017/S0960129516000268.

P. N. Benton. A mixed linear and non-linear logic: Proofs, terms and models. In Leszek Pacholski and Jerzy Tiuryn, editors, *Computer Science Logic*, pages 121–135, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg. ISBN 978-3-540-49404-1.

Aleš Bizjak and Lars Birkedal. On models of higher-order separation logic. *Electronic Notes in Theoretical Computer Science*, 336:57–78, 2018. ISSN 1571-0661. doi: https://doi.org/10.1016/j.entcs.2018.03.016. URL `https://www.sciencedirect.com/science/article/pii/S1571066118300197`. The Thirty-third Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXIII).

Peter Freyd. Algebraically complete categories. In Aurelio Carboni, Maria Cristina Pedicchio, and Guiseppe Rosolini, editors, *Category Theory*, pages 95–104, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg. ISBN 978-3-540-46435-8.

Daniel Gratzer, Evan Cavallo, G. A. Kavvos, Adrien Guatto, and Lars Birkedal. Modalities and parametric adjoints. *ACM Trans. Comput. Logic*, 23(3), apr 2022. ISSN 1529-3785. doi: 10.1145/3514241. URL `https://doi.org/10.1145/3514241`.

Claudio Hermida, Uday S. Reddy, and Edmund P. Robinson. Logical relations and parametricity – a reynolds programme for category theory and programming languages. *Electronic Notes in Theoretical Computer Science*, 303:149–180, 2014. ISSN 1571-0661. doi: https://doi.org/10.1016/j.entcs.2014.02.008. URL `https://www.sciencedirect.com/science/article/pii/S1571066114000346`. Proceedings of the Workshop on Algebra, Coalgebra and Topology (WACT 2013).

Paul-André Melliès. Categorical semantics of linear logic. 2009.

Eugenio Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, 1991. ISSN 0890-5401. doi: https://doi.org/10.1016/0890-5401(91)90052-4. URL `https://www.sciencedirect.com/science/article/pii/0890540191900524`. Selections from 1989 IEEE Symposium on Logic in Computer Science.

Max S. New and Daniel R. Licata. Call-by-name gradual type theory. In *FSCD 2018*, 2018.

Max S. New, Daniel R. Licata, and Amal Ahmed. Gradual type theory. In *POPL 2019*, 2019.

David J. Pym, Peter W. O'Hearn, and Hongseok Yang. Possible worlds and resources: the semantics of bi. *Theoretical Computer Science*, 315(1):257–305, 2004. ISSN 0304-3975. doi: https://doi.org/10.1016/j.tcs.2003.11.020. URL `https://www.sciencedirect.com/science/article/pii/S0304397503006248`. Mathematical Foundations of Programming Semantics.

R. A. G. Seely. Locally cartesian closed categories and type theory. *Mathematical Proceedings of the Cambridge Philosophical Society*, 95(1):33–48, 1984. doi: 10.1017/S0305004100061284.

Michael Shulman. Categorical logic from a categorical point of view. 2016. URL `https://mikeshulman.github.io/catlog/catlog.pdf`.

Philip Wadler. Monads for functional programming. In Manfred Broy, editor, *Program Design Calculi*, pages 233–264, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg. ISBN 978-3-662-02880-3.