Synthetic topology in Homotopy Type Theory for probabilistic programming

Florian Faissole
Inria
LRI, Université Paris-Sud & CNRS
Université Paris-Saclay
florian.faissole@lri.fr

Bas Spitters *
Aarhus University
spitters@cs.au.dk

1. Introduction

In the theory of programming languages, the use of proof assistants has become mainstream. It is considered good form to provide a formal connection between a language and its semantics. Currently, the main tools for this are based on either higher order logic, or on type theory. Here we will focus on Coq, the biggest system of the latter class. The ALEA [2] Coq library formalizes discrete measure theory using a variant of the Giry monad, as a submonad of the CPS monad: $(A \to [0,1]) \to [0,1]$. This allows to use Moggi's monadic meta-language to give an interpretation of a language Rml into type theory. Rml is a functional language with a primitive for probabilistic choice [2]. This formalization was the basis for the Certicrypt system [5] to verify security protocols. Easycrypt is still based on the same idea. To be precise, the type of the monad M of measures also requires monotonicity, summability and linearity. However, Coq cannot prove this to be a monad, as the equality on distributions is not the intensional equality of Coq. We solve this technical issue by using homotopy type theory. Moreover, this allows us to use synthetic topology to present a theory which also includes continuous datatypes like $[0,1]$. Such datatypes are relevant, for instance, in machine learning and differential privacy.

2. Probability theory and synthetic topology

The Giry monad [13] can be constructed on various categories, for instance on the category of Polish spaces, or on a category of domains [17]. We could first formalize these categories in Coq and then build the model on top of that. Such an approach is taken in Isabelle/HOL; e.g. [15]. However, we can stay closer to the ALEA library by using synthetic topology. Synthetic topology follows the philosophy of synthetic domain theory [16], a successful tool in semantics. In synthetic topology, one uses a domain specific language for topology, or more precisely, we interpret some language into a category which is sufficiently like that of topological spaces; see [3]. We use a topos with a special object Σ which classifies the (enumerable) opens. I.e. open subsets are replaced by maps to Σ. One may think of Σ as the Sierpinski space, since in Top there is a bijective correspondence between open subsets and the inverse image of a continuous map at $1 \in \Sigma$. In synthetic computability, one would take Σ to be the semi-decidable truth values. We emphasize that this use of synthetic topology is a mere convenient abstraction of standard realizability presentations of computations with continuous datatypes, such as Kleene's second algebra, or domain theory.

We assume two axioms from synthetic topology: Let $N^\bullet$ be the type of increasing binary sequences, 'the one-point compactification of N'. We assume:
WSO ('Weakly Sequentially Open'): The intrinsic topology, $N^\bullet \to \Sigma$, coincides with the metric topology $d(\underline{n}, \underline{m}) = 2^{-\min(n,m)}$. **WSO** holds in models of synthetic topology. **WSO** contradicts full classical logic, but combines well with exclude middle for types with decidable equality, which are heavily used in ssreflect [14]. Decidable properties are reflected by maps into the booleans. **WSO** fits with the interpretation where we reflect semi-decidable properties into Σ.

To define the measure on $[0,1]$ we will use: **Fan**: $2^{\mathbb{N}}$ is metrizable and compact. This axiom is necessary to avoid the singular covers of $[0,1]$ in a recursive interpretation of synthetic topology.

3. Homotopy type theory and univalent foundations

Coq's type theory lacks quotients and functional extensionality. To address this ALEA uses so-called setoids, a type together with an equivalence relation. This makes the library quite heavy since one needs to prove that all functions actually preserve this relation. Even though there is better support for this has been developed [22], there is now a more principled solution. Homotopy type theory [24] provides a consistent way of adding such features while conjecturally preserving the good computational properties. We use the HoTT library [4] for Coq which adds these features axiomatically. However, an experimental type checker for HoTT is already available [8] and we hope for its integration in proof assistants in the future. On top of HoTT, we add the axioms for synthetic topology. More precisely, we add them only for the so-called hSets. NuPrl [20] provides an extensional type theory which supports these axioms. However, we prefer Coq, as it is a more mature system and we also get some benefits from homotopy type theory, as discussed below.

Strictly speaking the models for synthetic topology have not been extended to type theory. However, sheaf models can be extended to models of homotopy type theory with so-called weak Tarski universes. One may also suspect that realizability models can be extended to homotopy type theory. Instead of trying to solve all the technical issues, we investigate whether this approach is useful; Shulman [21] takes a similar attitude.

Implementation in HoTT In comparison with NuPrl, HoTT gives us a few benefits. For instance, the univalence axiom is well-suited for algebraic and categorical reasoning [24]. Moreover, it facilitates the formalization of free (algebraic) structures. For instance, the

* This is an updated version of abstracts accepted for both CoqPL'16 and PPS'16. We have tried to answer the questions by the referees.
This research was partially supported by the Guarded homotopy type theory project, funded by the Villum Foundation, project number 12386 and Digiteo.

partiality monad [1] is the free ω-cpo completion, a quotient inductive inductive type (QIIT). We define a type $A_\perp$ with constructors, $\eta, \perp, \bigcup$ and a relation $\subseteq$ satisfying the expected relations.

$$A_\perp : hSet \qquad \subseteq_{A_\perp} : A_\perp \to A_\perp \to Type.$$
$$\eta : A \to A_\perp \qquad \perp : A_\perp$$
$$\bigcup : \prod_{f:\mathbb{N}\to A_\perp} (\prod_{n:\mathbb{N}} f(n) \subseteq_{A_\perp} f(n+1)) \to A_\perp$$

We set $\mathbb{S} := Unit_\perp$. By **WSO**, $X \to \mathbb{S}$ behaves like the open sets.

The Cauchy and Dedekind reals have been formalized in HoTT [12] based on an adaptation of the MathClasses library [18]. MathClasses provides an abstract approach to continuous computation, using type classes. On top of this, we use the lower reals, $\mathbb{R}_l$. These are lower (open) cuts in the rational numbers. Maps $X \to \mathbb{R}_l$ correspond to lower semi-continuous functions in synthetic topology. Similarly, we can define the upper reals. A consistent pair of an upper and a lower real defines a Dedekind real. From these we can define valuations and integrals on $A : hSet$:

Valuations:
$Val(A) = (A \to \mathbb{S}) \to [0,1]_l$

- $\mu(\emptyset) = 0$
- Modularity
- Monotonicity
- Continuity

Integrals:
$Int^+(A) = (A \to \mathbb{R}^+_D) \to \mathbb{R}^+_D$

- $\int \lambda\_.0 = 0$
- Additivity
- Monotonicity
- Probability: $\int \lambda\_.1 = 1$

We have a constructive Riesz theorem [9]: a homeomorphism between integrals and valuations for compact regular locales. This will allow us to develop a good constructive probability theory for spaces including $[0,1]$. However, to obtain a monad on all hSets, we need to consider lower integrals $Int^+_l(A) = (A \to \mathbb{R}^+_l) \to \mathbb{R}^+_l$. Vickers [25] proves Riesz' theorem: a homeomorphism between lower integrals and valuations, for all locales. We carry out a similar construction in synthetic topology and obtain a commutative monad on hSet:

- unit: $\eta_x(u) := \delta_x(u)$
- bind:

$$\mu_\varphi(u) := \int_{s \in M(X)} s(u)\, d\varphi(s).$$

The Dirac δ-function reduces to $d_x(u) := u(x)$, since $u : A \to \Sigma$.

4. Probabilistic languages

Like in ALEA, we can now use the monad to interpret Rml using Moggi's computational λ-calculus. In fact, we have an even richer type system since hSet is not only Cartesian closed, by functional extensionality, but even locally Cartesian closed, i.e. we have Π-types. Moreover, since the Kleisli category is ω-cpo enriched (we use subprobability valuations), we can interpret fixed points as in [2].

Non-measurable functions One may wonder how we avoid the classical problems of non-measurable functions. Classically $\Sigma = 2$, so a valuation on $[0,1]$ assigns a measure to *all* its subsets. This contradicts the axioms of choice. Fortunately, continuously, the elements of $[0,1] \to \Sigma$ are precisely the opens, hence we can model the Lebesgue valuation — the uniform distribution on $[0,1]$.

On the other hand, valuations on, for instance, compact Hausdorff spaces, are in bijective correspondence with regular measures. Hence, we capture one of the standard categories of spaces for the Giry monad. This puts our work in the *structural* approach to probability theory. We have a very good category of probability spaces, including the unit interval, while avoiding set theoretic anomalies. We already know from easycrypt, which can be modelled in our development, that much of probabilistic reasoning can be captured in such a way. Likewise, Panangadan [19] argues that in computer science one is interested in the structural properties of the category of stochastic relations build from the Giry monad.

Presheaves There is an interesting analogy with the semantics for higher order probabilistic programming in [23]. They first consider a fairly standard model for first-order probabilistic computation, say, the Giry monad on standard Borel spaces. To model function types, they use a variant of the Yoneda embedding.

A similar problem exists in synthetic topology, the category Top is not Cartesian closed. A common solution is to consider a convenient super-category. Escardo [10, Ch10] mentioned a number of subcategories of presheaves over Top for this purpose. In our case, it is more natural to consider the *sheaves* for the open cover topology and, in fact, we could take some gross topos on a topological site [11]. In this light, one could consider our construction as first completing with function types and then defining the monad on the bigger category.

5. Computability

In our formalization in Coq, we have used axioms from both synthetic topology and homotopy type theory. This means that we no longer have a guarantee that our evaluation terminates in Coq. However, there are implementations of these axioms in NuPrl [20] and cubical [8], respectively. Moreover, it is reasonable to expect that these features can be combined. One approach[1] implements the cubical model in NuPrl. An alternative would be to add the theory of names and effects from NuPrl to the cubical proof assistant in a way similar to the addition of guarded recursion to cubical [7].

The computational results one would obtain in such a framework are similar to the ones in ALEA. Since our language has general fixed points we cannot expect the semantics to terminate in general. However, the semantics will be semi-decidable. If a program p contains randomness from, say, only the unit interval, then the semantics is a valuation on the unit interval. Hence, we obtain a program which can semi-decide questions of the form $[[p]](I) > r$, where I is a rational interval in $[0,1]$ and r is a rational number.

In case we limit recursion *and* restrict to a class of 'compact regular' types one may expect a stronger result when integrating a (continuous) function with respect to the measure $[[p]]$, since in that case, the value of an integral is a Dedekind real, not just a lower real.

6. Conclusions and future work

We have combined homotopy type theory and synthetic topology to provide a new axiomatic semantics for probabilistic computation. This simplifies the ALEA library by the use of quotients and functional extensionality from HoTT and allows the addition of continuous data types. Our main insight is the extension of the Giry monad from locales to synthetic topology.

We have checked most of the details of the construction informally and hope to have a full formalization[2] soon. Presently, we have some 1500LOC consisting of the main constructions and definitions. For instance, we have a theory of the lower and upper reals and definitions of integrals and valuations. We have formalized the ω-cpo structure on the lower reals and valuations. Based on previous porting experience in the HoTT library, we expect to be able to port the discrete parts of the ALEA library, e.g. binomial coefficients.

There is a lot of active research on sheaf and realizability models for HoTT. However, the precise connection between this and the

[1] `http://www.math.ias.edu/vladimir/files/Bickford_Slides.pdf`

[2] `https://github.com/FFaissole/Valuations/`

implementation in Coq is still open; see also [4].
ALEA provides axiomatic semantics for Rml, a similar approach works in our case. It would also be interesting to deeply embed Rml into Coq. This would make it possible to connect an operational and the denotational semantics. In [6] it is argued that, unlike in higher order logic, in type theory one can directly define a dependently typed map from syntax to semantics and that this is important for the verification of, e.g. compiler optimizations.

Acknowledgments

The questions in this paper originated from discussions with Christine Paulin in 2014, when Spitters had a Digiteo chair at LRI, Inria. We also benefited from Faissole's internship with Paulin on formalizing the lower reals in Coq. We are grateful for both.

We thank the referees for their questions and suggestions.

References

[1] T. Altenkirch, N. A. Danielsson, and N. Kraus. Partiality, Revisited: The Partiality Monad as a Quotient Inductive-Inductive Type. *ArXiv:1610.09254*, 2016.

[2] P. Audebaud and C. Paulin-Mohring. Proofs of randomized algorithms in Coq. In *MPC*, 2006.

[3] A. Bauer and D. Lesnik. Metric spaces in synthetic topology. *Ann. Pure Appl. Logic*, 163:87–100, 2012.

[4] A. Bauer, J. Gross, P. L. Lumsdaine, M. Shulman, M. Sozeau, and B. Spitters. The HoTT library. *arXiv:1610.04591*, 2016.

[5] S. Z. Béguelin. Formal certification of game-based cryptographic proofs. (certification formelle de preuves cryptographiques basées sur les séquences de jeux). 2010.

[6] N. Benton, L. Birkedal, A. Kennedy, and C. Varming. Formalizing domains, ultrametric spaces and semantics of programming languages. 2010.

[7] L. Birkedal, A. s Bizjak, R. Clouston, H. B. Grathwohl, B. Spitters, and A. Vezzosi. Guarded Cubical Type Theory. *ArXiv:1611.09263*, 2016.

[8] C. Cohen, T. Coquand, S. Huber, and A. Mörtberg. Cubical type theory: a constructive interpretation of the univalence axiom. *Proc. Types for Proofs and Programs (TYPES 2015)*, 2016. URL `https://www.math.ias.edu/~amortberg/papers/cubicaltt.pdf`.

[9] T. Coquand and B. Spitters. Integrals and valuations. *Journal of Logic and Analysis*, 1(3):1–22, 2009. ISSN 1759-9008. doi: 10.4115/jla.2009.1.3.

[10] M. Escardó. Synthetic topology: of data types and classical spaces. *ENTCS*, 87:21–156, 2004.

[11] M. Fourman. Continuous truth II: Reflections. In *WoLLIC*, 2013.

[12] G. Gilbert. Formalising real numbers in homotopy type theory. *arXiv:1610.05072*, 2016.

[13] M. Giry. A categorical approach to probability theory. In *Categorical aspects of topology and analysis*, pages 68–85. 1982.

[14] G. Gonthier and A. Mahboubi. An introduction to small scale reflection in Coq. *Journal of Formalized Reasoning*, 3(2):95–152, 2010. URL `https://hal.inria.fr/inria-00515548`.

[15] J. Hölzl, A. Lochbihler, and D. Traytel. A formalized hierarchy of probabilistic system types - proof pearl. In C. Urban and X. Zhang, editors, *ITP*, volume 9236 of *LNCS*, pages 203–220. Springer, 2015.

[16] J. M. E. Hyland. First steps in synthetic domain theory. In *Category Theory*, pages 131–156. Springer, 1991.

[17] C. Jones and G. D. Plotkin. A probabilistic powerdomain of evaluations. In *LICS*, 1989.

[18] R. Krebbers and B. Spitters. Type classes for efficient exact real arithmetic in Coq. *LMCS*, 9(1:1):1–27, 2013. doi: 10.2168/LMCS-9(1:01)2013.

[19] P. Panangaden. Probabilistic relations. In *School of Computer Science, McGill University, Montreal*, pages 59–74, 1998.

[20] V. Rahli and M. Bickford. A nominal exploration of intuitionism. In *CPP*, 2016.

[21] M. Shulman. Brouwer's fixed-point theorem in real-cohesive homotopy type theory. *arXiv:1509.07584*, 2015.

[22] M. Sozeau. A new look at generalized rewriting in type theory. *Journal of Formalized Reasoning*, 2(1):41–62, 2010.

[23] S. Staton, H. Yang, C. Heunen, O. Kammar, and F. Wood. Semantics for probabilistic programming: higher-order functions, continuous distributions, and soft constraints. *CoRR*, abs/1601.04943, 2016.

[24] T. Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations for Mathematics*. `http://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.

[25] S. Vickers. A monad of valuation locales. 2011. URL `http://www.cs.bham.ac.uk/sjv/Riesz.pdf`.