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 system of the latter class. The ALEA [2] Coq library formalizes discrete measure theory using a variant of the Giry monad, as a system of the latter class. The ALEA [2] Coq library formalizes discrete measure theory using a variant of the Giry monad, as a system of the latter class. The ALEA [2] Coq library formalizes discrete measure theory using a variant of the Giry monad, as a system of the latter class.

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 system of the latter class. The ALEA [2] Coq library formalizes discrete measure theory using a variant of the Giry monad, as a system of the latter class. The ALEA [2] Coq library formalizes discrete measure theory using a variant of the Giry monad, as a system of the latter class. The ALEA [2] Coq library formalizes discrete measure theory using a variant of the Giry monad, as a system of the latter class.

We assume two axioms from synthetic topology: Let \( N^* \) be the space of increasing binary sequences, the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( \Sigma \) 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 \( N^* \rightarrow \Sigma \) be the space of increasing binary sequences, the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( N^* \rightarrow \Sigma \) be the one-point compactification of \( N \). We assume the axiom of synthetic topology: Let \( \Sigma \) 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.

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

\[
A_\bot : h\text{Set} \\
\eta : A \to A_\bot \\
\bot : A_\bot \\
\bigcup : \prod_{f : n \to A_\bot} f(n) \subseteq A_\bot \to A_\bot
\]

We set \( S := Unit_\bot \). By WSO, \( X \to S \) 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 \( A : h\text{Set} \):

\[
Val(A) = (A \to S) \to [0, 1]; \\
Int^+(A) = (A \to \mathbb{R}_l^+ \to \mathbb{R}_l^+)
\]

- \( \mu(\emptyset) = 0 \)
- \( \int \lambda \cdot 0 = 0 \)
- Modularity
- Additivity
- Monotonicity
- Monotonicity
- Continuity
- Probability: \( \int \lambda \cdot 1 = 1 \)

We have a constructive Riesz theorem [9]: a homeomorphism between intervals and valuations for compact regular locales. This will allow us to develop a good constructive probability theory for continuous data types. Our main insight is the extension of 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 \( \text{the sheaves} \) 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 \( \langle [p] \rangle(I) > r \), 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 \( \langle [p] \rangle \), 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 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