Towards a Formalization of Affine Schemes in Cubical Agda

Anders Mörtberg and Max Zeuner

Stockholm University, Stockholm, Sweden
{anders.mortberg,zeuner}@math.su.se

Schemes are the corner stone of modern algebraic geometry. Roughly speaking, they are topological spaces equipped with a sheaf of rings that locally look like so-called *affine schemes*. These affine schemes arise from central notions of commutative algebra: the *Zariski topology* defined on the set of prime ideals of a commutative ring R is equipped with a particular sheaf of rings, the *structure sheaf* using *localizations* of R. An early formalization of affine schemes in the `Coq` proof assistant can already be found in [4] but more recently, a full-blown formalization of general schemes was added to `Lean`'s `mathlib` [3]. By now, schemes have also been defined in `Isabelle/HOL` [2] and in `Coq`'s `UniMath`-library.[1] All of the aforementioned formalizations follow the inherently non-constructive approach of Hartshorne's classic textbook "Algebraic Geometry" [7] when defining the structure sheaf.

Working in `Cubical Agda`, an extension of the `Agda` proof assistant based on cubical type theory with fully constructive support of the univalence axiom [11], we want to give a *constructive* formalization of affine schemes in a univalent setting. This also seems to be in line with the aim of Voevodsky's Foundations library [12] to develop a library of constructive set-level mathematics based on a univalent foundation. To this end, we decided to follow the approach described by Coquand et. al. in [5]. This can be seen as a constructivization of the common approach that first defines the structure sheaf on the *basic opens*, a canonical basis of the Zariski topology, and then use some general machinery to extend this to a sheaf on the whole space.

Localization Localizations are generalized fractions and for the structure sheaf we need localizations at a single element: for a ring R and an element $f \in R$, the localization $R[1/f]$ corresponds roughly to the ring of fractions of the form x/f^n. Localizations can directly be defined as set-quotients, one of `Cubical Agda`'s higher inductive types (HIT). The universal property of localization gives us for two elements $f, g \in R$ a unique isomorphism of rings $R[1/f][1/g] \cong R[1/fg]$. When verifying the sheaf property of the structure sheaf, these two rings are however identified and freely substituted across the proof in informal mathematics. Using univalence, or more precisely `Cubical Agda`'s structure identity principle [1], we can promote this isomorphism to an equality $R[1/f][1/g] = R[1/fg]$. At this point our formalization differs already from the `Lean` formalization. Without univalence this equality is not obtainable, which led the authors of [3] to adapt a somewhat non-standard approach to localization.

Zariski Lattice The main difference between the constructive approach of [5] and classical ones is the use of a synthetic description of the Zariski lattice. We have to give a point-free definition of this lattice since prime ideals do not behave well constructively. Classically, open sets of the Zariski topology are generated by basic opens $D(f)$, the set of prime ideals of R that do *not* contain $f \in R$. The synthetic Zarsiki lattice is the free distributive lattice generated by *formal symbols* $D(f)$ quotiented by some relations that make these formal basic opens behave like their classical counterparts.

This synthetic definition was first given by Joyal in [8], but in our formalization we use a more explicit construction due to Español [6]: The Zariski lattice is formalized as `List` R, the

[1] See https://github.com/UniMath/UniMath/tree/master/UniMath/AlgebraicGeometry

type of lists with elements in R, quotiented by the relation

$$[\alpha_0, \ldots, \alpha_n] \sim [\beta_0, \ldots, \beta_m] \quad :\Leftrightarrow \quad \sqrt{\langle \alpha_0, \ldots, \alpha_n \rangle} = \sqrt{\langle \beta_0, \ldots, \beta_m \rangle}$$

This means that two lists are considered equal if the *radicals* of the ideals generated by their respective elements are equal. Since we are thus only working over finitely generated ideals, elements of this quotient are in bijective correspondence with open sets of the Zariski topology if R is *Noetherian*.

Equipping this set-quotient with the structure of a distributive lattice requires a lot of standard results about radical ideals and facts about $\_ + \_$ and $\_ \cdot \_$ of ideals, as those will give rise to the lattice operations. On top of that we need to introduce operations $\_ ++ \_$ and $\_ \cdot\cdot \_$ on lists that correspond to addition and multiplication of the ideals generated by those lists. For addition $\_ ++ \_$ is just list concatenation and for multiplication $\alpha \cdot\cdot \beta$ is the list of all products of the form $\alpha_i \beta_j$. The bottleneck then becomes proving that

$$\begin{aligned} \langle [\alpha_0, \ldots, \alpha_n] ++ [\beta_0, \ldots, \beta_m] \rangle &= \langle \alpha_0, \ldots, \alpha_n \rangle + \langle \beta_0, \ldots, \beta_m \rangle \\ \langle [\alpha_0, \ldots, \alpha_n] \cdot\cdot [\beta_0, \ldots, \beta_m] \rangle &= \langle \alpha_0, \ldots, \alpha_n \rangle \cdot \langle \beta_0, \ldots, \beta_m \rangle \end{aligned}$$

Structure Sheaf In the above implementation of the Zariski lattice, the basic open $D(f)$ corresponds to the equivalence class of the singleton list $[f]$. In HoTT/UF [10], subsets of a type X are functions from X into hProp, the universe of (homotopy) propositions. So if we denote by L_R the Zariski lattice associated with R, the basic opens are a function $B_R : L_R \to \mathsf{hProp}$ mapping $\alpha \in L_R$ to

$$B_R(\alpha) \quad := \quad \exists_{f:R} \left(D(f) = \alpha \right) \quad := \quad \| \Sigma_{f:R} \left(D(f) = \alpha \right) \|$$

Here $\|\_\|$ is the propositional truncation, another HIT in `Cubical Agda`, turning any type into a proposition. The general strategy is now to construct the structure sheaf on basic opens and then use some general results from category theory to obtain a sheaf on the whole Zariski lattice. The construction of the presheaf on basic opens as well as a weak form of the sheaf property, are fully formalized.[2] The full sheaf property and its lift to the whole lattice are currently work in progress.

For defining the basic presheaf we need to give a map $\left(\Sigma_{\alpha:L_R} B_R(\alpha)\right) \to \mathsf{CommRing}$ from elements of the Zariski lattice that are basic opens into the type of commutative rings, mapping $D(f)$ to $R[1/f]$. Here we run into a problem resulting from working in a univalent setting: Given an α s.t. $B_R(\alpha)$, the information which $D(f)$ equals to α is hidden under a propositional truncation while the goal type $\mathsf{CommRing}$ is a *groupoid* (i.e. a 1-type). To eliminate this truncation we need to appeal to a result by Kraus [9]. This however requires us to check some higher coherence condition, i.e. construct square fillers in $\mathsf{CommRing}$. The key observation at this point is that the structure sheaf on L_R does actually take values in R-Alg, the type of R-algebras, as localizations of R are always R-algebras. The ring-valued structure sheaf can then be obtained by composing with the forgetful functor R-$\mathsf{Alg} \to \mathsf{CommRing}$. Some standard commutative algebra tells us that for $f, g \in R$ with $\sqrt{\langle f \rangle} \subseteq \sqrt{\langle g \rangle}$ there is a *unique* homomorphism of R-algebras $R[1/g] \to R[1/f]$.

It turns out that this not only solves the coherence issues, but actually facilitates the presheaf construction and the proof of the (weak) sheaf property by a rather neat argument that makes essential use of the primitives of `Cubical Agda` as well as the equality $R[1/f][1/g] = R[1/fg]$, while avoiding cumbersome diagram chases.

[2] See https://github.com/agda/cubical/tree/master/Cubical/Algebra/ZariskiLattice

References

[1] Carlo Angiuli, Evan Cavallo, Anders Mörtberg, and Max Zeuner. Internalizing representation independence with univalence. *Proc. ACM Program. Lang.*, 5(POPL), January 2021.

[2] Anthony Bordg, Lawrence Paulson, and Wenda Li. Simple type theory is not too simple: Grothendieck's schemes without dependent types. arXiv preprint, 2021. https://arxiv.org/abs/2104.09366.

[3] Kevin Buzzard, Chris Hughes, Kenny Lau, Amelia Livingston, Ramon Fernández Mir, and Scott Morrison. Schemes in lean. *Experimental Mathematics*, 0(0):1–9, 2021.

[4] Laurent Chicli. Une formalisation des faisceaux et des schémas affines en théorie des types avec Coq. Technical Report RR-4216, INRIA, June 2001.

[5] Thierry Coquand, Henri Lombardi, and Peter Schuster. Spectral schemes as ringed lattices. *Annals of Mathematics and Artificial Intelligence*, 56(3):339–360, 2009.

[6] Luis Español. Le spectre d'un anneau dans l'algèbre constructive et applications à la dimension. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 24(2):133–144, 1983.

[7] Robin Hartshorne. *Algebraic geometry*, volume 52. Springer Science & Business Media, 2013.

[8] André Joyal. Les théoremes de chevalley-tarski et remarques sur l'algèbre constructive. *Cahiers Topologie Géom. Différentielle*, 16:256–258, 1976.

[9] Nicolai Kraus. The general universal property of the propositional truncation. In Hugo Herbelin, Pierre Letouzey, and Matthieu Sozeau, editors, *20th International Conference on Types for Proofs and Programs (TYPES 2014)*, volume 39 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 111–145, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[10] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. https://homotopytypetheory.org/book, Institute for Advanced Study, 2013.

[11] Andrea Vezzosi, Anders Mörtberg, and Andreas Abel. Cubical agda: A dependently typed programming language with univalence and higher inductive types. *Journal of Functional Programming*, 31:e8, 2021.

[12] Vladimir Voevodsky. An experimental library of formalized mathematics based on the univalent foundations. *Mathematical Structures in Computer Science*, 25(5):1278–1294, 2015.