A computational logic (coinductive) interface for schemes in algebraic geometry.

[updated] https://github.com/1337777/cartier/blob/master/cartierSolution16.lp

This is the continuation of an ongoing research programme of discovering a truly computational logic (type theory) for categories, profunctors, fibred categories, univalence, polynomial functors, sites, sheaves and schemes, in the style of Kosta Dosen.

A glue operation for sheaves `R` over the sheafification modality `mod_smod` is declared:

```
    constant symbol glue : Π [A B : cat] (A_site : site A)  [R :
smod A_site B] [I : cat] [G : func I B] (L : mod A I),
transf L Id_func (smod_mod R) G
→  transf (smod_mod (mod_smod A_site L)) Id_func (smod_mod R) G;
```

From which a glue operation for sheaves `R` over the sieve-closure modality `sieve_ssieve` (not endotransformation …) is defined, where `sieve` is the presheaf (profunctor …) which classifies sieves:

```
    symbol glue_sieve_mod_def : Π [A B : cat] (A_site : site A) [I
: cat] [F : func I A]  [R : smod A_site B]  [G : func I B] [D :
cat] [K : func I D] (ff : hom F (sieve A D) K),
transf (sieve_mod ff) Id_func (smod_mod R) G)
→  transf (smod_mod (ssieve_smod ((ff '∘ (sieve_ssieve A_site _)))))
Id_func (smod_mod R) G ≔ glue …;
```

And when this sieve-closure happens to become the maximal sieve, then this glue operation is indeed the reverse operation of the (new …) Yoneda action by sieve-elements. In short: Nicolas Tabareau [1] is only a formalization of the semantics of sheafification, not an actual computational logic; and moreover, its Definition 5.2 causes flaws: instead of “a subobject of E is dense in E if …”, it should be “a subobject P of E is dense in another subobject Q of E if …”.

There are also (substructural) versions of this story in the presence of context: tensor-product context A⊗B ⊢ C, subtype context Σ(a:A),P(a) ⊢ C, and dependent-type context (x:A)|B(x) ⊢ C(x).

Warning to users: it is not necessary to use multi-categories to manage contexts, don’t be indoctrinated by Mikey [2]. Why? Because a question which is never entertained by logicians is:

“Who is my end-user?”

The answer lies in the elimination rule for the tensor product: if a:A, b:B ⊢ C then A⊗B ⊢ C. Each time that this rule would be used, instead its intention can be simulated by the LAMBDAPI end-user by manually (via macros …) adding a rewrite rule which outputs the intended body content of A⊗B ⊢ C when applied to a constructor ⟨a, b⟩. In short: multi-categories are for the dummy end-user.

Now the presentation of affine schemes is such that it exposes a logical interface/specification which computes; for example the structure sheaf `ascheme_ring_loc` localized away from r : R has restrictions along D(s·r) ⊆ D(r) (and along explicit radicals D(r) ⊆ D(r^n) …) which compute:

```
    rule (ascheme_mult_hom $R $s $r) '∘ (_) _'∘> (ring_loc_intro
(ascheme_ring_loc $R $r) $x)
↪ ring_loc_intro (ascheme_ring_loc $R (ring_mult $R $s $r))
(ring_mult $R $s $x) ;
```

And similar computations happen for formal joins D(f)∨D(g) of basic opens. But Joyal's covering axiom D(s+r) ⊆ D(s)∪D(r) of the basic open D(s+r), has been reformulated approximately as a cover by the inclusions D(s·√(a·s + b·r)) ⊆ D(√(a·s + b·r)) (similarly for r) such to handle also the unimodular cover of a basic open.

Finally, there is an attempt to present locally ringed sites and schemes. But the traditional definition of schemes via isomorphisms to affine schemes won't work computationally; instead, one has to declare that the slice-category sites of the base site satisfy the interface/specification of an affine scheme. But what is a slice-category site? How does its glue operation relate to the glue operation of the base site? This has required subtle reformulations of a cocontinuous-and-continuous morphism of sites with a continuous right adjoint [3].

But then, what is the invertibility support D(–) for a locally ringed site and how does it relate to each affine-scheme's invertibility support D(–) in the slice-categories. The author of cartierSolution16.lp claims that D(f) is intended to be the SIEVE (possibly singleton …) under U of all opens V where the function f: O(U) becomes invertible, together with (a computational reformulation of) the condition:

```
    lim_{V:D(f)} O(V)  =  O(U)[1/f]
```

In short, some structured data is being transferred from a base scheme to its slice-categories where they are required to satisfy the affine-scheme interface. Moreover, the affine-scheme interface is coinductive (self-reference), meaning that its slice-categories are also required to satisfy the affine-scheme interface. A consequence of this formulation is that these various structure sheaves are now canonically related, beyond the mere usual knowledge that:

```
    R[1/fg]  ≅   R[1/f][1/g]
```

MAX [4] recently defended an equivalence between functorial schemes X and locally-ringed-lattice schemes Y, approximately:

```
    LRDL^op( X ⇒ Spec , Y )  ≅   ( X ⇒ LRDL^op( Spec( – ), Y ) )
```

where (X ⇒ X') ≔ Fun(CommRing , Set)(X , X') and Spec: Fun(

CommRing^op, LRDL^op). But the author of cartierSolution16.lp claims that their profunctor framework should allow a hybrid handling of schemes as locally ringed sites together with their functor-of-points semantics. And most importantly, it is not necessary to try and express those things using an internal logic (not truly-computational …) within the Zarisky topos, in the style of Thierry Coquand [5].

Ultimately, it is a conjecture that this new framework could solve the open problem of discovering a (graded) differential linear logic formulation for the algebraic-geometry's cohomology differentials via the profunctorial semantics of linear logic in the context of sieves as profunctors…

It is also a conjecture that this new framework could solve the search of a hybrid framework· combining polynomial functors (good algebra) "depending" on analytic functors (good logic) as motivated by Ehrhard's [6, section 3.1.1, page 44] outrageous definition of the composition of polynomials by the use of differentials instead of by elementary algebra.

Another open question: would such computational-logical interfaces for commutative (affine) schemes be able to also specify schemes of (noncommutative) associative algebras; for example, in the sense of Siqveland Arvid [7]?

[1] Nicolas Tabareau. "Lawvere-Tierney sheafification in homotopy type theory"
[2] Mikey. "Categorical logic from a categorical point of view (for dummies)"
[3] The Stacks Project. "Section 7.22: Cocontinuous functors which have a right adjoint"
[4] MAX Zeuner. "Univalent Foundations of Constructive Algebraic Geometry"
[5] Thierry Coquand. "A foundation for synthetic algebraic geometry"
[6] Thomas Ehrhard. "An introduction to differential linear logic: proof-nets, models and antiderivatives"
[7] Siqveland Arvid. "Schemes of Associative algebras"

"Bla bla bla, PhD, etc…" OK, but why don't YOU invest to pay a "PhD review" of those documents. As the circulation of math reviews becomes a currency, the word qualifiers for "theft", "falsification", "intoxication", "assault" should no longer be limited to the traditional context of a cash-bank-robbery-by-a-gang.

Wait, before engaging in smart astronauts' arguments in the vacuum of the galaxy, maybe we should come down to Earth with fresh air which can propagate the sound waves of those discussions. Indeed, the author has already anticipated such vacuum arguments, by inserting into the transcript record this 1+2=3 sanity-check challenge ("contexte et préalables d'un débat") against ∞-cosmonauts, lol …

https://github.com/1337777/cartier/blob/master/cartierSolution14.lp

https://github.com/1337777/cartier/blob/master/Kosta_Dosen_2pages.pdf

where the author shows that 1+2=3 via 3 different methods: the category of natural-numbers as a higher inductive type; the natural-numbers object inside any fixed category; and the colimits inside the category of finite sets/numbers.

Indeed, this new functorial programming language, also referred as Dosen’s « m— » or « emdash » or « modos », is able to express the usual logic such as the tensor and internal-hom of profunctors, the sigma-sum and pi-product of fibred categories/profunctors; but is also able to express the concrete and inductively-constructed categories/profunctors, to express the adjunctions such as the product-and-exponential or the constant-diagram-and-limit adjunctions within any fixed category, to express contravariance and duality such as a computational-proof that right-adjoints preserve limits from the dual statement, to express groupoids and univalent universes, to express polynomial functors as bicomodules in the double category of categories, functors, cofunctors and profunctors, etc…

Such an ongoing multi-year research programme requires new tools and frameworks to truly enable the circulation of (math) reviews as a currency; for example, new platforms such as this author's « re365.net » open-source Microsoft 365 app, developed by a community (https://meetup.com/dubai-ai) of 2,000+ contributors, which is a tiktok-style AI-assisted calendar overlay for all such communications between researchers/businesses and reviewers whose side-goals are to co-author or by-product more intelligent (AI) interfaces for their papers/apps API. (https://dailyReviews.link)