Workshop on implementing Kosta Dosen's categorial programming and sheaves: Cut-elimination in the double category of profunctors with J-rule-eliminated adjunctions.

Short: There is now sufficient evidence (ref [6], [7]) that Kosta Dosen's ideas and techniques (ref [1], [2], [3], [4], [5]) could be implemented for proof-assistants, sheaves and applications; in particular cut-elimination, rewriting and confluence for various enriched, internal, indexed or double categories with adjunctions, monads, negation, quantifiers or additive biproducts; quantitative/quantum linear algebra semantics; presheaf/profunctor semantics; inductive-sheafification and sheaf semantics; sheaf cohomology and duality... It was difficult to discover the correct grammatical-formulation out of a dozen semantically-meaningful ones, but this ongoing implementation (ref [6]) should work in the next months.

The problem of "formulations of adjunction", the problem of "unit objects" and also the problem of "contextual composition/cut" can be understood as the same problem. The question arises when one attempts to write precisely the counit/eval transformation ϵ_X ： catA(F G X, X) where F : catB ⟶ catA is the left-adjoint. One could instead write ϵ_X ： catA[F ,1](G X, X) where the profunctor object/datatype catA[F ,1] is used in lieu of the unit hom-profunctor catA; in other words: F is now some implicit context, and the contextual composition/cut (in contravariant action) of some g : catB[1, G] (Y, X) in the unit profunctor against ϵ_X should produce an element of the same datatype: ϵ_X◦g : catA[F,1](Y,X)... Ultimately Dosen-Petric (ref [5]) would be extended in this setting where some dagger compact closed double category, of left-adjoint profunctors across Cauchy-complete categories, is both inner and outer dagger compact closed where the dagger operation on profunctors (as 1-cells) coincides with the negation operation on profunctors (as 0-cells), optionally with sheaf semantics and cohomology.

Some oversight about the problem of "unit objects" is the belief that it should have any definitive once-for-all solution. Instead, this appears to be a collection of substructural problems for each domain-specific language. Really, even the initial key insight of Dosen-Petric, about the cut-elimination formulation in the domain-specific language of categorial adjunctions, can be understood as a problem of "unit profunctor" in the double category of profunctors and the (inner) cut elimination lemma becomes synonymous with elimination of the "J-rule"; and recall that such equality/path-induction J-rule would remain stuck in (non-domain-specific) directed (homotopy) type theory.

Now the many "formulations of adjunctions" are for different purposes. Indeed the outer framework (closed monoidal category, with conjunction bifunctor ∧ with right adjoint implication connective ⇒) hosting such inner domain-specific language (unit-counit formulation of adjunctions) could itself be in another new formulation of adjunctions (lambda/eval bijection of hom-sets) where the implication bifunctor (with also contravariant argument _⇒_) is accumulated during computation via dinaturality (in contrast to the traditional Kelly-MacLane formulation).

This is the insight that leads Kosta Dosen to say that any ordinary natural transformation

$$t_A : F\ G\ A \to H\ K\ A$$

can be formulated as an "*antecedental transformation*"

$$\frac{f : K\ A \to B}{Hf \circ t_A : F\ G\ A \to H\ B}$$

with primitive name "$H - \circ\ t$" in the language, or can be formulated as a "*consequential transformation*"

$$\frac{f : B \to G\ A}{t_A \circ Ff\ : F\ B \to H\ K\ A}$$

with primitive name "$t\ \circ\ F -$" in the language. And in the special case when t_A is the counit of an adjunction with the functor F left adjoint to G and with H, K absent ($H = 1, K = 1$), then these various formulations allow for the elimination/admissibility of the composition ∘ (cut-elimination). Of course, this cut elimination is except those (apparent) cuts baked into the primitive language of the antecedental or consequential formulation of the counit or unit; nevertheless, the decidability of the equality of the arrows still holds via the confluence lemma.

For reference (§ 4.1.5 in [1]), some Dosen-style adjunction with left adjoint functor $F: catB \to catA$, right adjoint functor $G: catA \to catB$, counit transformation $\phi_A: F\ G\ A \to A$, and unit transformation $\gamma_B: B \to G\ F\ B$ is formulated as rewrite rules from any redex outer cut on the left-side to the contractum containing some smaller inner cut:

$$f_2 \circ \text{“}f_1 \circ \phi\text{”} = \text{“}(f_2 \circ f_1) \circ \phi\text{”}$$

$$\text{“}\gamma \circ g_2\text{”} \circ g_1 = \text{“}\gamma \circ (g_2 \circ g_1)\text{”}$$

$$“f \circ \phi” \circ “F(g)” = f \circ “\phi \circ Fg”$$

$$“G(f)” \circ “\gamma \circ g” = “Gf \circ \gamma” \circ g$$

together with conversion (or rewrite) rules:

$$“\phi \circ F(“Gg \circ \gamma”)” = g$$

$$“G(“\phi \circ Ff”) \circ \gamma” = f$$

where indeed the functions on arrows $F -$ and $G -$ of those functors are not primitive but are themselves the “consequential transformation” formulations “$1 \circ F -$” of the identity natural transformation…

New functorial lambda calculus. Now such Dosen-style technique may be specialized to the instance of closed monoidal categories where the conjunction bifunctor _⋀_ has right adjoint implication bifunctor _⇒_ via lambda/eval bijection of hom-sets. Then dinaturality is used to accumulate the argument-component of the eval operation instead on its function-component:

$$“\epsilon_{B,O} \circ B \wedge (g)” \circ (f \wedge x)$$

$$= “\epsilon_{A,O} \circ A \wedge \big((x \Rightarrow O) \circ (g \circ f)\big)”, \qquad x: A \to B$$

where this evaluation rule uses skewed tensors, which are disguised generalized Kan extensions:

| Eval_cov_transf : Π [A B C X A' X' : Cat] [P : mod A B] [Q : mod C X] [O : mod A' X'] [K : func B C] [F : func A A'] [L : func X X'],
P ⊢ F ◦> (Imply_cov_mod (O <◦ L) Q) <◦ K ⟶ (Tensor_cov_mod P (K ◦> Q)) ⊢ F ◦> O <◦ L

In fact, such antecedental/consequential transformations may be formulated systematically, similarly as the “J-rule” for equality/paths in (homotopy) type theory, if general bimodule/profunctor-homs ($HomP : catA^{op} \times catB \to Set$) such as ($F\!-\!\!\to -$) or ($-\!\!\to G-$) are expressible in the language. _And most importantly, because the general “J-rule” hide some cuts, therefore cut-elimination signify that the general “J-rule” must also be eliminated/admissible/computational._

New grammatical topology. A sheaf is data defined over some topology, and sheaf cohomology is linear algebra with data defined over some topology. A closer inspection reveals that there is some intermediate formulation which is computationally-better that Cech cohomology: at least for the standard simplexes (line, triangle, etc.), then intersections of opens could be internalized as primitive/generating opens for the cover and become points in the nerve of this cover (as suggested by the barycentric subdivision). This redundant storage space for functions defined over the topology is what allows possibly-incompatible functions to be glued, and to prove the acyclicity for the standard simplex (and to compute how this acyclicity fails in the presence of holes in the nerve). For example, the sheaf data type:

$$F(U0) \coloneqq \text{sum over the slice U0 U01} = \mathbb{Z} \oplus \mathbb{Z};$$

$$F(U1) \coloneqq \mathbb{Z} \oplus \mathbb{Z}; F(U01) \coloneqq \mathbb{Z}$$

$$F(U) = \text{kan extension} = \mathbb{Z} \oplus \mathbb{Z} \oplus \mathbb{Z};$$

gives the gluing operation

$$\text{gluing}: F(U0) \oplus F(U1) \oplus F(U01) \to F(U)$$

$$\big((f0, f01), (g1, g01), (h01)\big) \mapsto (f0, g1, f01 + g01 - h01)$$

where the signed sum generalizes to higher degrees because the Euler characteristic is 1.

The implementation would use some local operator $j: \Omega \to \Omega$ where $\Omega(A)$ is the classifier of (sub-)objects (sieves) of the object A, and where $j_A(\mathcal{U})(f) \coloneqq f^*\mathcal{U} \in J(X)$ is the (opaque) set of witnesses that the pullback-sieve $f^*\mathcal{U}$ is covering (remember that the truthness that $\mathcal{U}$ is covering is expressed as $\mathcal{U} \in J(A)$, iff $\forall f.\, j(\mathcal{U})(f)$). But it is better to consider any presheaf in the slice over A, rather than only subfunctors because then everything is expressible as profunctor-homs (of witnesses) over some slice categories. Now the “J-rule” would be formulated to allow as output some sheafification modality and as input any possibly-incompatible family over some covering sieve, not only the singleton family over the generating identity-arrow of the trivial cover...

References:

[1] Dosen-Petric: Cut Elimination in Categories 1999;
[2] Proof-Theoretical Coherence 2004;
[3] Proof-Net Categories 2005;
[4] Coherence in Linear Predicate Logic 2007;
[5] Coherence for closed categories with biproducts 2022
[6] Cut-elimination in the double category of profunctors with J-rule-eliminated adjunctions: https://github.com/1337777/cartier/blob/master/cartierSolution12.v (Ongoing) https://github.com/1337777/cartier/blob/master/cartierSolution12.lp (Ongoing, primary file)
[7] Pierre Cartier