Di- is for Directed: First-Order Directed Type Theory via Dinaturality

ANDREA LARETTO, FOSCO LOREGIAN, and NICCOLÒ VELTRI, TalTech, Estonia

We show how dinaturality plays a central role in the interpretation of directed type theory where types are given by (1-)categories and directed equality by hom-functors. We introduce a first-order directed type theory where types are semantically interpreted as categories, terms as functors, predicates as dipresheaves, and proof-relevant entailments as dinatural transformation. This type theory is equipped with an elimination principle for directed equality, motivated by dinaturality, which closely resembles the J-rule used in Martin-Löf type theory. This directed J-rule comes with a simple syntactic restriction which recovers all theorems about symmetric equality, except for symmetry. Dinaturality is used to prove properties about transitivity (composition), congruence (functoriality), and transport (coYoneda) in exactly the same way as in Martin-Löf type theory, and allows us to obtain an internal "naturality for free". We then argue that the quantifiers of directed type theory should be ends and coends, which dinaturality allows us to capture formally. Our type theory provides a formal treatment to (co)end calculus and Yoneda reductions, which we use to give distinctly logical proofs to the (co)Yoneda lemma, the adjointness property of Kan extensions via (co)ends, exponential objects of presheaves, and the Fubini rule for quantifier exchange. Our main theorems are formalized in Agda.

CCS Concepts: • **Theory of computation → Type theory**.

Additional Key Words and Phrases: directed type theory, coend calculus, dinaturality

ACM Reference Format:
Andrea Laretto, Fosco Loregian, and Niccolò Veltri. 2018. Di- is for Directed: First-Order Directed Type Theory via Dinaturality. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 42 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Homotopy type theory [7, 78, 81] revolutionized the way we think about types. One of the fundamental insights that inspired this revolution was first given in a seminal paper by Hofmann and Streicher [42], with a remarkably simple idea: rather than viewing types just as *sets* of inhabitants, they give an interpretation of Martin-Löf type theory where types are taken to be *groupoids*, i.e., categories in which every morphism is an isomorphism. The inhabitants of a type become the objects of a groupoid, and the morphisms in a groupoid represent the *equalities* between inhabitants, of which there can be more than a unique one. The reason why morphisms need to be invertible is because of the inherently *symmetric* nature of equality: given a proof of equality $e : x = y$, there is always a proof of the equality $e' : y = x$.

A natural question follows: why not *categories*, rather than groupoids? Can there be a type theory where types are interpreted as *categories*, where morphisms need not be invertible? Such a system should take the name of *directed type theory* [2, 4, 34, 51, 61, 84], where the directed aspect comes precisely from this asymmetric interpretation of "equality".

Authors' Contact Information: Andrea Laretto; Fosco Loregian; Niccolò Veltri, TalTech, Estonia.

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

Types C	Categories $\mathbb{C}$
Functions $f : C \to D$	Functors $F : \mathbb{C} \to \mathbb{D}$
Relations $R : C \times D \to \mathsf{Bool}$	Profunctors $P : \mathbb{C}^{\mathsf{op}} \times \mathbb{D} \to \mathsf{Set}$
Predicates $P : C \to \mathsf{Bool}$	Presheaves $P : \mathbb{C}^{\mathsf{op}} \to \mathsf{Set}$
Points of a type	Objects of a category
Equalities $e : a =_C b$	Morphisms $e : \hom_{\mathbb{C}}(a, b)$
Equality types $=_C : C \times C \to \mathsf{Type}$	Hom functors $\hom_{\mathbb{C}} : \mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathsf{Set}$
Universal quantifiers	Ends $\int_{x:\mathbb{C}} P(\overline{x}, x)$
Existential quantifiers	Coends $\int^{x:\mathbb{C}} P(\overline{x}, x)$

Fig. 1. The directed generalization of logical concepts.

Directed type theory has been a hot topic of type-theoretical research for the past decade [4, 19, 35, 59, 60, 62, 64, 83]. This quest for the directed generalization has a specific application in mind: in the same way that HoTT can be used to study homotopy theory in a type-theoretical way, directed type theory promises the study of *category theory* in a type-theoretical way.

Category theory has proven to be a fundamental topic in the semantics of programming languages [23, 46, 57, 76], where it shines as the common framework that ties together logic, proofs, and types in the Curry-Howard-Lambek correspondence [17, 37, 43]. The *unifying* role of category theory stretches even beyond computer science, in algebraic topology [53], universal algebra [47], quantum mechanics [39], and physics [8].

This compelling series of applications comes at a cost: category theory can be overwhelming for newcomers, with overly abstract results and seemingly complicated ideas (e.g., the Yoneda lemma [15], Kan extensions [40]). Even worse, these abstractions come baggaged with a plethora of naturality and functoriality side conditions that need to be checked [60].

Directed type theory promises to reinterpret category theory *itself* under a logical perspective, taking the Curry-Howard-Lambek correspondence to the next level: what once were abstract yet overarching results in category theory become *simple type-theoretical statements*, which one can then prove in a system that takes care of naturality and functoriality bureaucracy *for free*.

One of the ultimate goals of directed type theory is to capture this multitude of directed phenomena under a single, unified type-theoretical framework: since morphisms of a category can be viewed just as (directed) equalities, one can use directed type theory as a tool to represent and reason about programs, processes, rewrites, transitions [1], concurrency via directed spaces [28, 61], types and terms of type theories (e.g., via "directed higher inductive types" [44, 83]), *all internally* to the same type theory.

What is currently missing from the current conception of directed type theory is a direct description of what such a system should look like in the elementary case of 1-categories. Taking inspiration from the simplicity of the groupoid model in Hofmann and Streicher's approach,

We introduce a first-order directed type theory with simple, straightforward semantics in 1-categories: proving theorems about directed equality follows the same exact steps of Martin-Löf type theory, and non-trivial theorems in category theory can be captured in a concise and distinctly logical way.

How should type-theoretical ideas change under the view of directed type theory? Category theorists have long known what the most natural path for the directed generalization should be [49]: functions between types should be *functors* (i.e., functions which respect directed equalities), relations are naturally interpreted as *profunctors* [16], and *(co)presheaves* can be thought of as generalized predicates [9]. We summarize the main ideas of the *directed generalization* in Figure 1.

Under this directed lens, familiar type-theoretical statements of equality become elementary definitions in category theory: we give a few simple examples in Figure 2 in the canonical setting of *first-order logic*, which is closely connected to the formal system later explored in this paper.

$x = y \quad \wedge \quad y = z \qquad \vdash \qquad x = z$ $\hom_{\mathbb{C}}(x, y) \times \hom_{\mathbb{C}}(y, z) \to \hom_{\mathbb{C}}(x, z)$	Transitivity of equality Composition in a category
$x = y \qquad \vdash \qquad f(x) = f(y)$ $\hom_{\mathbb{C}}(x, y) \to \hom_{\mathbb{D}}(F(x), F(y))$	Congruence / functions respect equality Action on morphisms of functors
$x = y \wedge P(x) \ \vdash \ P(y)$ $\hom_{\mathbb{C}}(x, y) \times P(x) \to P(y)$	Substitution / transport along equality Action on morphisms of copresheaves

Fig. 2. Elementary statements for symmetric equality and their directed counterparts.

However, directed type theory is not so straightforward. We list some fundamental challenges:

Challenge 1. How to change rules for equality. One can use their favorite proof assistant or logical system to prove the theorems in Figure 2: in the case of symmetric equality, typically this is done using an introduction rule (refl=) and an elimination rule (J=) called *J-rule* [41], shown in Figure 3 again for first-order logic. The introduction rule simply states that equality is reflexive. The elimination rule J intuitively says that, if we assume an equality $e : a = b$ and we want to prove a predicate $P(a, b)$ for some variables $a, b : C$, it is sufficient to consider the case "on the *diagonal*" $P(x, x)$, where a and b are identified with the same x. These two rules allow all of the above statements about symmetric equality to be derived almost "for free" just by contracting away equalities. However, (J=) allows for symmetry of equality to be derived, simply by picking $P(a, b) := b = a$. This is incompatible with the directed case, as not every morphism has an inverse.

The fundamental question then becomes: *how can we tweak the rules of equality to disallow symmetry, and yet be able to derive "for free" the above theorems also in the case of directed equality?*

$$\frac{}{[x : C] \ \Phi \vdash \mathsf{refl} : x = x} \ (\mathsf{refl}{=}) \qquad \frac{[x : C] \qquad \Phi(x, x) \vdash \qquad h : P(x, x)}{[a : C, b : C] \ a = b, \ \Phi(a, b) \vdash J(h) : P(a, b)} \ (J{=})$$

Fig. 3. Introduction and elimination rules for symmetric equality in first-order logic.

$$\frac{}{[x : \mathbb{C}] \ \Phi \vdash \mathsf{refl} : \hom_{\mathbb{C}}(\overline{x}, x)} \ (\mathsf{refl}) \qquad \frac{[x : \mathbb{C}] \qquad \Phi(x, \overline{x}) \vdash \qquad h : P(\overline{x}, x)}{[a : \mathbb{C}^{\mathrm{op}}, b : \mathbb{C}] \ \hom(a, b), \ \Phi(\overline{a}, \overline{b}) \vdash J(h) : P(a, b)} \ (J)$$

Fig. 4. Introduction and elimination rules for directed equality in first-order dinatural directed type theory.

Challenge 2. Polarity problems. Another issue arises in the first example of Figure 2: since types are now categories, with each type $\mathbb{C}$ there should be a type $\mathbb{C}^{\mathrm{op}}$ (the opposite category) of the opposite "polarity", where the inhabitants are the same but all directed equalities are reversed. The *type of directed equalities* $\hom_{\mathbb{C}}(x, y)$ then is *asymmetric*, and receives a "negative" argument $x : \mathbb{C}^{\mathrm{op}}$ and a "positive" one $y : \mathbb{C}$, and provides the *set* (i.e., a category with only trivial directed equalities) of morphisms between objects x, y of $\mathbb{C}$.

The problem is that in the statement for transitivity of directed equality (i.e. composition) the variable y appears both on the right side of $\hom_{\mathbb{C}}(x, y)$, with type $\mathbb{C}$, and at the same time on the left side of $\hom_{\mathbb{C}}(y, z)$, with seemingly different type $\mathbb{C}^{\mathrm{op}}$! The same problem arises in (refl), since x is used on both sides of hom, and in (J) because in $P(x,x)$ the same x needs to be used with both polarities. One solution first considered by North [61] and later revisited by Altenkirch and Neumann [4] is to revert back to the undirected case of *groupoids*. This solution may feel unsatisfactory, since one does not intuitively expect groupoids to appear in the semantics of a type theory where types are *categories*. *How do we solve these polarity problems without having to resort to groupoids?*

Challenge 3. Directed quantifiers. Another fundamental yet unexplored question is *what the quantifiers of directed type theory should be in the 1-categorical case*. Because of the above polarity issues, this is not a trivial question: should the variable y in the statement of transitivity be bound as a variable of type $y : \mathbb{C}$ or $\overline{y} : \mathbb{C}^{\mathrm{op}}$? A natural expectation is that quantifiers should be able to bind *both* occurrences of y at once.

This paper proposes a simple solution that addresses all of the above challenges for directed type theory: dinaturality [26].

The intuition behind dinaturality and dinatural transformations is that the same variable is allowed to appear both positively and negatively at the same time, irrespectively of polarity.

Not only do we deal with the variance problems without ever having to mention groupoids, but dinaturality also tells us what a *directed* J rule should look like, which we illustrate in Figure 4 next to the symmetric case. Curiously, this rule is reminiscent of the elimination rule for equality of standard Martin-Löf type theory, but it comes equipped with a precise syntactic restriction that does not allow symmetry of directed equality to be derived.

What about quantifiers? Dinaturality comes again to the rescue, hinting at a possible answer: intimately connected to the notion of dinatural transformation are the notions of *end* and *coend* [52]. Ends and coends, respectively denoted as $\int_{x:\mathbb{C}} P(\overline{x},x)$ and $\int^{x:\mathbb{C}} P(\overline{x},x)$ for some functor $P : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathsf{Set}$, are to be thought of as a sort of universal and existential quantifiers on P, respectively. Just like a quantifier, the integral sign of (co)ends binds positive and negative occurrences of variables, indicated as $x : \mathbb{C}$ and $\overline{x} : \mathbb{C}^{\mathrm{op}}$.

The main application of (co)ends is that they allow non-trivial statements in category theory to be formulated in a concise way [52]: for example, one can use ends to characterize the set of natural transformations as the end $\mathsf{Nat}(F,G) \cong \int_{x:\mathbb{C}} \hom_{\mathbb{D}}(F(\overline{x}),G(x))$; note the resemblance between this end and the universal quantification used in the usual definition of natural transformation. With this, we can rephrase the well-known Yoneda lemma [50] as a simple isomorphism, shown in Figure 5a next to its logical "decategorified" interpretation. A similar statement holds for the case of existential quantifiers and coends, shown in Figure 5b, which often takes the slogan of "presheaves are colimits of representables" [50] or "coYoneda lemma" [20, 52].

$$\text{(a)}\ \frac{P(a) \ \cong\ \int_{x:\mathbb{C}} \ \hom_{\mathbb{C}}(a,\overline{x}) \Rightarrow P(x)}{P(a) \Leftrightarrow \forall (x:C).\ \ a =_C x \ \ \Rightarrow P(x)} \qquad \text{(b)}\ \frac{P(a) \ \cong\ \int^{x:\mathbb{C}} \ \hom_{\mathbb{C}}(\overline{x},a) \times P(x)}{P(a) \Leftrightarrow \exists (x:C).\ \ x =_C a \ \ \wedge P(x)}$$

Fig. 5. Yoneda and coYoneda lemma using (co)ends and their corresponding logical statements.

The first-order formula behind the (co)Yoneda lemma can be proven using any formal system: our directed type theory is the first elementary treatment of a formal system for the *directed* case, where one can modularly use rules for quantifiers and equality as done in logic, e.g., with suitable introduction/elimination rules specific to directed equality and (co)ends. To give a taste of how closely our approach follows that of a standard logical proof, we show in Figure 6 a proof of the Yoneda lemma in our type theory next to its "decategorified" proof in first-order logic.

$$
\begin{array}{c}
[a{:}C]\ \Phi(a) \vdash \forall(x:C).\ a =_C x \Rightarrow P(x) \\
\hline\hline
[a{:}C, x{:}C]\ \Phi(a) \vdash a =_C x \Rightarrow P(x) \\
\hline\hline
[a{:}C, x{:}C]\ \ a =_C x \wedge \Phi(a) \ \vdash P(x) \\
\hline\hline
[z:C]\ \Phi(z) \vdash P(z)
\end{array}
\begin{array}{l}
(\forall) \\ (\Rightarrow) \\ (=)
\end{array}
\qquad
\begin{array}{c}
[a{:}\mathbb{C}]\ \Phi(a) \vdash \int_{x:\mathbb{C}} \hom_{\mathbb{C}}(a, \overline{x}) \Rightarrow P(x) \\
\hline\hline
[a{:}\mathbb{C}, x{:}\mathbb{C}]\ \Phi(a) \vdash \hom_{\mathbb{C}}(a, \overline{x}) \Rightarrow P(x) \\
\hline\hline
[a{:}\mathbb{C}, x{:}\mathbb{C}]\ \ \hom_{\mathbb{C}}(\overline{a}, x) \times \Phi(a) \ \vdash P(x) \\
\hline\hline
[z:\mathbb{C}]\ \Phi(z) \vdash P(z)
\end{array}
\begin{array}{l}
(\text{end}) \\ (\text{exp}) \\ (J)
\end{array}
$$

Fig. 6. A proof of the Yoneda lemma in first-order logic, and its proof in dinatural directed type theory.

(Co)end calculus. It is common knowledge among category theorists that there is a formal aspect to the manipulation of ends and coends, outlined in [52], that allows such non-trivial theorems to be proven using simple "mechanical" rules. This "(co)end calculus" has proven to be particularly useful in theoretical computer science, for example in the context of profunctor optics [15, 20] and their string diagrams [14, 74], strong monads and functional programming [5, 6, 40, 80], quantum circuits [38], and logic [31, 68, 70]. Our work gives a *logical interpretation* to (co)end calculus by reconceptualizing it just as a first-order instance of directed type theory, which is what motivates our focus on a non-dependent presentation of directed type theory.

Dinaturality. Dinaturality is not a novel concept: dinatural transformations are a generalization of natural transformations for functors $F, G : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{D}$ with mixed-variances [26].

Serendipitously, the "*di*" in dinatural stands for *diagonal*: a dinatural is a family of maps $\alpha_x : F(x, x) \to G(x, x)$ which is required to be given only on the *diagonal* of F, G by equating the contravariant and covariant variables with the same value $x : \mathbb{C}$. Such family of maps is required to satisfy a certain equational property, which generalizes the usual square of natural transformations.

Famously, however, dinatural transformations *do not always compose*: a well-known sufficient condition for the composability of dinaturals is the absence of loops in a suitably associated graph [27, 55]. This loop-freeness similarly arises in linear logic with the Danos-Regnier criterion [11–13, 36], and more in general in logic where composition corresponds to cut elimination [32, 66].

There is a particularly deep connection between dinaturality and parametricity in programming languages [67, 69, 72, 82] and realizable models for System F [10, 29] where all dinaturals compose. Dinaturality has remained somewhat of an understudied subject, partly because this lack of general compositionality has proven to be particularly hard to explain in full generality [75]: an in-depth review on dinaturality and its importance for computer science can be found in [75], [76, Sec. 3].

1.1 Contribution

In this work, we connect for the first time dinatural transformations to directed type theory, showing how they turn out to be the key technical notion needed to capture directed type theory in an elementary and straightforward way.

Our general approach to directed type theory is to take the simplicity of the groupoid model of Hofmann and Streicher [42] and generalize it to the directed case with a first-order (yet expressive) system aimed at capturing two specific aspects of directed type theory: first, the ability to construct and prove properties about theorems of directed equality by following precisely the same steps as in Martin-Löf type theory; second, the ability to exploit the power of (co)ends-as-quantifiers [52] to give simple and concise logical proofs of well-known theorems in category theory.

We summarize the main contributions and technical aspects of this paper:

(1) *Setting*. We introduce a first-order (non-dependent) directed type theory where types are semantically interpreted as (small) 1-categories, terms as functors, predicates as dipresheaves (i.e. functors $\mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathrm{Set}$), directed equality predicates as hom-functors, and proof-relevant entailments as dinatural transformations which are not required to compose in the usual sense.

(2) *First-order type theory.* Our directed type theory builds on the well-known canonical setting of first-order logic, with judgments structured in a similar way [43, 4.1]: we have simply-typed types and terms, on which we build a proof-relevant logic with predicates, entailments, and *equality of entailments.* This last aspect is typically absent in usual accounts of first-order logic, but it is crucial in our presentation because it is precisely the point in which we use dinaturality. Our system is a *type theory* in the sense of Jacobs [43, p. 9, (iii)]: proofs have an explicit computational content, e.g., the proof of transitivity of directed equality is a bona-fide family of functions that can be used to compose equality witnesses (i.e., morphisms) in the type theory.

(3) *Directed equality elimination.* In our 1-categorical setting, the rules for directed equality are straightforward: the directed equality introduction rule is essentially the same as the usual refl, which we validate using identities in hom-sets. We identify a directed equality elimination rule which is again syntactically reminiscent of the J-rule, but equipped with a syntactic restriction that does not allow for symmetry to be derived. This syntactic restriction is not ad-hoc, but it is justified by a precise semantic fact: the connection between dinaturality and ordinary naturality. In short, the syntactic requirement to contract a directed equality in context $\hom_{\mathbb{C}}(x, y)$ for $x : \mathbb{C}^{\mathrm{op}}, y : \mathbb{C}$ is that both x and y must appear only covariantly (i.e., with the "correct polarity") in the conclusion and only contravariantly (i.e., with the "wrong polarity") in the assumptions in context. The non-derivability of symmetry, aside from the syntactic restriction of J, follows by soundness and the existence of a countermodel.

(4) *Directed theorems.* The rules for directed equality allow us to recover in Section 3 the same type-theoretic definitions about symmetric equality derivable in standard Martin-Löf type theory, except for symmetry: e.g., transitivity of directed equality (composition in a category), congruences of terms along directed equalities (the action of a functor on morphisms), transport along directed equalities (i.e., the coYoneda lemma).

(5) *Directed properties.* In our type theory one can also prove *properties* of these maps using a *dependent* version of directed J specific to the judgment of equality of entailments: for example, one can show that the composition of directed equalities is automatically associative and unital on both sides (one of the two sides is definitionally unital on the equality that is being contracted). The semantic notion of dinaturality is not used to *construct* such maps (functoriality suffices), but to validate this dependent directed J rule. With this rule one can internally prove that functoriality and naturality follow "for free", again, by a simple directed equality contraction.

(6) *Polarity.* Our type theory is equipped with a precise notion of polarity and variance which is used to implement the syntactic restriction behind the J rule. Even in our non-dependent case the treatment of variables is non-trivial, since dinaturality requires a precise definition of variance/polarity that differs from the approaches described in other works [4, 34, 61, 63].

(7) *Category theory, logically.* Our type theory allows us to give direct, concise, and distinctly logical proofs of well-known (yet non-trivial) theorems in category theory by using hom as a directed equality: e.g., the (co)Yoneda lemma, Kan extensions computed via (co)ends are adjoint to precomposition, presheaves form a closed category, hom-functors preserve (co)limits, and the Fubini rules; each of these follows by modularly using the logical properties of each connective.

(8) *(Co)end calculus.* The approach used to prove these theorems is to combine the perspective of hom as directed equality with the ideas of "(co)end calculus" [52], viewing (co)ends as the *directed quantifiers* of directed type theory. (Co)end calculus as treated in [52] uses various semantic properties of (co)ends, which are however selected *ad-hoc* and not modularly organized in a precise set of rules: our type theory gives a formal treatment to these techniques, approaching

proofs in a different and more logical fashion. The choice of a first-order (hence non-dependent) type theory is to capture (co)end calculus, which is typically first-order in practical applications.

(9) *Yoneda technique.* Our proofs are logical, yet mirror the way that (co)end calculus is used in practice (e.g., [15, 40, 74]), i.e., via a "Yoneda-like" series of *natural* isomorphisms of sets: to prove that two objects $A, B : \mathbb{C}$ are isomorphic, one assumes a generic object Φ and then applies a series of isomorphisms of sets *natural* in Φ to establish that $\mathbb{C}(\Phi, A) \cong \mathbb{C}(\Phi, B)$, from which $A \cong B$ follows by the fully faithfulness of the Yoneda embedding [15, 50]. The same technique can be used to show adjunctions, and that *functors* are naturally isomorphic.

(10) *Adjoint-form rules.* In typical syntactic presentations of type theory, rules for connectives are formulated to make cut admissible [41, 77]. In our case, we cannot have in the semantics that all entailments (i.e. dinaturals) compose, and therefore our rules must be stated in such a way that cut is not admissible. In his seminal paper [48], Lawvere introduced the categorical understanding of logic by viewing quantifiers/connectives as adjoints: we formulate (some of) the rules of our type theory with dinaturals precisely in Lawvere's "adjoint-form" (e.g. [43, 4.1.7, 4.1.8]), i.e., as natural bijections between entailments. In standard accounts of logic this adjoint-form is equivalent to the usual intro/elim. rules for connectives, but only in the presence of *cut*; the key observation is that, despite the absence of a general cut rule, the rules for quantifiers/connectives in adjoint-form *can* be validated in our semantics with dinaturals.

(11) *(Co)ends-as-quantifiers.* The rules for ends and coends are reminiscent of the quantifiers-as-adjoints paradigm by Lawvere [48], which we captured as "right and left adjoint" operations to weakening [43, 1.9.1]. This adjointness relation should be only interpreted suggestively, since (co)ends are functorial operations for naturals but in general not dinaturals [52, 1.1.7]. Our approach has the advantage that several properties of quantifiers, e.g., that they can be exchanged and permuted, follow automatically from certain *structural properties of contexts*. For example, in first-order logic the formulas $\forall x.\forall y.P \Leftrightarrow \forall y.\forall x.P \Leftrightarrow \forall (x, y).P$ are logically equivalent for any predicate P: this is indeed also verified for ends (and coends with existentials), and takes the name of "Fubini rule" [53, IX.8], [52, 1.3.1], which we prove in Example 6.4. More details on (co)ends and their calculus can be found in [53, IX.5-6], [52, Ch. 1].

(12) *Dinaturality.* Dinatural transformations do not compose in general [75]: this lack of general composition turns out not to be a problem in practice, since they *do* compose in all examples of interest. In such cases, dinaturals compose essentially because they compose with other *natural* transformations [26], and we capture this in our system by providing two *restricted* cut rules.

Because of the lack of general compositionality, we do not consider a categorical semantics of our type theory using standard categorical models, e.g., fibrations [43] or categories with families [18], since they all ask for full composition, which cannot be guaranteed in our semantics. Hence, our approach is to simply consider the main rules described in Figure 11 (which have *restricted* rules for composition of entailments) and prove soundness w.r.t. the category model with dinaturals.

We formalize the soundness theorems given in this paper about dinaturality using the Agda proof assistant and the `agda-categories` library. Whenever present, the symbol ( ) next to theorems links to the formal proof, for which we report here just the core idea. The full formalization is accessible at `https://github.com/iwilare/dinaturality`.

1.2 Related work

Directed type theory has been approached in several (mutually incompatible) ways, with different methodological choices regarding semantics and rules for directed equality, but without ever investigating the connection to dinaturality.

Directed type theory with groupoids. North [61], Altenrkirch and Neumann [4] describe a dependent directed type theory with semantics in the category of (small) categories Cat, but using groupoidal structure to deal with the problem of variance in both introduction and elimination rules for directed equality. This line of research has been recently expanded in [19, 62] by extending judgments with variance annotations.

We focus on non-dependent semantics, and avoid groupoids by tackling the variance issue with dinatural transformations; using dinaturality and (co)ends-as-quantifiers allow us to capture naturality for free and characterize natural transformations inside of the type theory.

Directed type theory, judgmental models. Another approach to modeling directed equality is at the judgmental level. This line of research started with Licata and Harper [51] who introduced a directed type theory with a model in Cat. Since directed equality is treated judgmentally, there are no rules governing its behavior in terms of elimination and introduction principles, although variances are similarly used in context as in our approach. Ahrens et al. [2] similarly identify a type theory with judgmental directed equalities and semantics in comprehension bicategories, and extensively compare previous works on both judgmental and propositional directed type theories.

Logics for category theory. New and Licata [60] give a sound and complete presentation for the internal language of (hyperdoctrines of) certain virtual equipments. These models capture enriched, internal, and fibered categories, and have an intrinsically directed flavor. In these contexts, the type theory can give synthetic proofs of Fubini, Yoneda, and Kan extensions as adjoints. This generality however comes at the cost of a non-standard syntactic structure of the logic, especially when compared to standard Martin-Löf type theory. Directed equality elimination takes the shape of the (horizontal) identity laws axiomatized in virtual equipments [24], which in the Prof model is essentially the coYoneda lemma. Their quantifiers are given by tensors and (left/right) internal homs, which in Prof correspond to certain restricted (co)ends which always come combined with the tensors and internal homs of Set.

Our work is similar in spirit in that we provide a formal setting for proving category theoretical theorems using logical methods; we only focus on the elementary 1-categorical model of categories and do not yet capture enriched and internal settings. However, we treat (co)ends as quantifiers *directly*, viewing them as operations which act on the variables of the context, without the need for them to include any conjunction or implication. Our rules for directed equality are more direct and reminiscent of standard Martin-Löf type theory, and specifically focus on the semantic justification of dinaturality. Since we consider less general models, our contexts do not have any linear nor ordered restriction and the same variable can appear multiple times both in equalities and contexts: for example, this allows us to *write down* the statement of symmetry (without being able to prove it), and to consider profunctors of arbitrary variables, as typically needed in (co)end calculus.

Geometric models of directed type theory. Riehl and Shulman [73] introduce a simplicial type theory for synthetic $(\infty, 1)$-categories. A directed interval type is axiomatized in a style reminiscent of cubical type theory [22], which allows a form of (dependent) Yoneda lemma to be derived using such identity type. This type theory has been implemented in practice in the Rzk proof assistant [45]. On this line of research, Weaver and Licata [84] present a *bicubical* type theory with a directed interval and investigate a directed analog of the univalence axiom; the same objectives were recently advanced in Gratzer et al. [34, 35] with triangulated type theory and modalities.

In comparison with the above works, we do not explore the geometrical interpretation of directedness and focus on "algebraic" 1-categorical semantics; moreover, our treatment of directed equality is done intrinsically with elimination rules as in Martin-Löf type theory rather than with synthetic intervals, with semantics directly provided by hom-functors.

Coend calculus, formally. Caccamo and Winskel [25] give a formal system in which one can work with coends and establish non-trivial theorems with a few syntactical rules. The flavor is explicitly that of an axiomatic system, and does not take inspiration from type-theoretic rules: for instance, presheaves are *postulated* to be equivalent under the swapping of quantifiers (Fubini), so this principle is not derived from structural rules as typically done in a logical presentation.

1.3 Structure of the paper

We start in Section 2 by describing syntax and judgmental structure of the type theory, and give examples of directed type theory in Section 3. We recall notions about dinaturals in Section 4, which we then use for the semantics in Section 5. We then apply our type theory to give logical proofs of theorems in category theory in Section 6, concluding in Section 7 with future works.

2 Syntax

We introduce the main syntactic judgments of our proof-relevant first-order directed type theory, for which we describe the main ideas and notation in Sections 2.1 and 2.2.

Our type theory is structured in a similar way to first-order logic [43, 4.1], with judgments for types and terms (i.e., sorts and function symbols), and predicates indexed by a term context.

We will omit several uninteresting equality judgments for contexts, terms, propositional contexts, as well as usual congruence and equivalence rules. We list here the main judgments of our type theory alongside a brief description of their semantics to aid intuition, with details in Section 5.

Figure 7:
- $\boxed{\mathbb{C}\ \mathsf{type}}$ **types** $\mathbb{C}, \mathbb{D}$ are interpreted in the semantics as small categories. Types can have $-^{\mathsf{op}}$, and include the terminal $\top$, product $\mathbb{C} \times \mathbb{D}$, and functor categories $[\mathbb{C}, \mathbb{D}]$.
- $\boxed{\mathbb{C} = \mathbb{D}}$ **judgmental equality of types**, interpreted as isomorphisms of categories; we use this to simplify $(\mathbb{C}^{\mathsf{op}})^{\mathsf{op}} = \mathbb{C}$ and propagate the op inside types.

Figure 8:
- $\boxed{\Gamma\ \mathsf{ctx}}$ **contexts** Γ, Δ are finite lists of categories, interpreted as *products in* Cat*s*;
- $\boxed{\Gamma \ni x : \mathbb{C}}$ **variable in context**, which captures the de Bruijn indices of variables in context Γ; for us variable names are irrelevant, and we always identify variables with these judgments. Semantically, these are the projections out of $[\![\Gamma]\!]$.
- $\boxed{\Gamma \vdash F : \mathbb{C}}$ **terms** F, G as *functors* $[\![\Gamma]\!] \to [\![\mathbb{C}]\!]$, which are similar to terms in STLC;

Figure 9:
- $\boxed{[\Gamma]\ P\ \mathsf{prop}}$ **predicates** P, Q as *dipresheaves*, i.e., functors $[\![P]\!] : [\![\Gamma]\!]^{\mathsf{op}} \times [\![\Gamma]\!] \to \mathsf{Set}$;
- $\boxed{[\Gamma]\ \Phi\ \mathsf{propctx}}$ **propositional contexts** Φ, Φ' are finite lists of predicates, which we interpret via the *pointwise product of dipresheaves in* Set;

Figure 11:
- $\boxed{[\Gamma]\ \Phi \vdash \alpha : P}$ **entailments** α, β, γ are interpreted semantically as *dinatural transformations* $[\![\Phi]\!] \xrightarrow{\bullet\bullet} [\![P]\!]$; we axiomatize composition/cut only with *natural* transformations, not requiring general composition;
- $\boxed{[\Gamma]\ \Phi \vdash \alpha = \beta : P}$ **equality of entailments**, i.e. *equality of dinaturals in* Set.

For predicates we consider the following logical connectives, which we denote syntactically with the same symbol later used in the semantics:

- **conjunction** $- \times -$, interpreted as the pointwise product of dipresheaves in Set;
- **polarized implication** $- \Rightarrow -$, by postcomposing dipresheaves with $\hom_{\mathsf{Set}} : \mathsf{Set}^{\mathsf{op}} \times \mathsf{Set} \to \mathsf{Set}$;
- **propositional directed equality** $\hom_{\mathbb{C}}$ is interpreted by hom-*functors* : $\mathbb{C}^{\mathsf{op}} \times \mathbb{C} \to \mathsf{Set}$;
- **universal and existential quantifiers** $\int_{x:\mathbb{C}} P(\overline{x}, x)$, $\int^{x:\mathbb{C}} P(\overline{x}, x)$ are given by *ends* and *coends*.

The judgments for types, terms, propositions and entailments are given in Figures 7 to 9 and 11.

Our directed type theory is equipped with an equational theory for entailments, which we describe the key features of in Section 2.2 without spelling it out in detail. The most important cases are given in Figure 11 for directed equality, Figure 15 for cuts, Figure 16 for adjoint rules.

$$\boxed{\mathbb{C}\ \mathsf{type}} \quad \frac{C \in \Sigma_B}{C\ \mathsf{type}} \quad \frac{\mathbb{C}\ \mathsf{type}}{\mathbb{C}^{\mathsf{op}}\ \mathsf{type}} \quad \frac{\mathbb{C}\ \mathsf{type} \qquad \mathbb{D}\ \mathsf{type}}{\mathbb{C} \times \mathbb{D}\ \mathsf{type}} \quad \frac{\mathbb{C}\ \mathsf{type} \qquad \mathbb{D}\ \mathsf{type}}{[\mathbb{C}, \mathbb{D}]\ \mathsf{type}} \quad \frac{}{\top\ \mathsf{type}}$$

$$\boxed{\mathbb{C} = \mathbb{D}} \quad \frac{}{(\mathbb{C}^{\mathsf{op}})^{\mathsf{op}} = \mathbb{C}} \quad \frac{}{(\mathbb{C} \times \mathbb{D})^{\mathsf{op}} = \mathbb{C}^{\mathsf{op}} \times \mathbb{D}^{\mathsf{op}}} \quad \frac{}{[\mathbb{C}, \mathbb{D}]^{\mathsf{op}} = [\mathbb{C}^{\mathsf{op}}, \mathbb{D}^{\mathsf{op}}]} \quad \frac{}{\top^{\mathsf{op}} = \top} \quad \cdots$$

Fig. 7. Syntax of first-order dinatural directed type theory – types and judgmental equality.

$$\boxed{\Gamma\ \mathsf{ctx}} \quad \frac{}{[]\ \mathsf{ctx}} \quad \frac{\Gamma\ \mathsf{ctx} \qquad \mathbb{C}\ \mathsf{type}}{\Gamma, \mathbb{C}\ \mathsf{ctx}} \quad \frac{\Gamma\ \mathsf{ctx}}{\Gamma^{\mathsf{op}}\ \mathsf{ctx}}$$

$$\boxed{\Gamma = \Gamma'} \quad \frac{}{[]^{\mathsf{op}} = []} \quad \frac{}{(\Gamma, \mathbb{C})^{\mathsf{op}} = \Gamma^{\mathsf{op}}, \mathbb{C}^{\mathsf{op}}} \quad \frac{\mathbb{C} = \mathbb{C}' \qquad \Gamma = \Gamma'}{\Gamma, \mathbb{C} = \Gamma', \mathbb{C}'}$$

$$\boxed{\Gamma \ni x : \mathbb{C}} \quad \frac{}{\Gamma, x : \mathbb{C} \ni x : \mathbb{C}} \quad \frac{\Gamma \ni x : \mathbb{C}}{\Gamma, y : \mathbb{D} \ni x : \mathbb{C}}$$

$$\boxed{\Gamma \vdash t : \mathbb{C}} \quad \frac{\Gamma \ni x : \mathbb{C}}{\Gamma \vdash x : \mathbb{C}} \quad \frac{\Gamma \vdash t : \mathbb{C}}{\Gamma^{\mathsf{op}} \vdash t^{\mathsf{op}} : \mathbb{C}^{\mathsf{op}}} \quad \frac{f \in \Sigma_T \qquad \Gamma \vdash t : \mathsf{dom}(f)}{\Gamma \vdash f(t) : \mathsf{cod}(f)}$$

$$\frac{}{\Gamma \vdash\, ! : \top} \quad \frac{\Gamma \vdash s : \mathbb{C} \qquad \Gamma \vdash t : \mathbb{D}}{\Gamma \vdash \langle s, t \rangle : \mathbb{C} \times \mathbb{D}} \quad \frac{\Gamma \vdash p : \mathbb{C} \times \mathbb{D}}{\Gamma \vdash \pi_1(p) : \mathbb{C}} \quad \frac{\Gamma \vdash p : \mathbb{C} \times \mathbb{D}}{\Gamma \vdash \pi_2(p) : \mathbb{D}}$$

$$\frac{\Gamma \vdash s : [\mathbb{C}, \mathbb{D}] \qquad \Gamma \vdash t : \mathbb{C}}{\Gamma \vdash s \cdot t : \mathbb{D}} \quad \frac{\Gamma, x : \mathbb{C} \vdash t(x) : \mathbb{D}}{\Gamma \vdash \lambda x.t(x) : [\mathbb{C}, \mathbb{D}]}$$

$$\boxed{\Gamma \vdash t = t' : \mathbb{C}} \quad \frac{\Gamma, x : \mathbb{C} \vdash f(x) : \mathbb{D} \qquad \Gamma \vdash t : \mathbb{C}}{\Gamma \vdash (\lambda x.f(x)) \cdot t = f[x \mapsto t] : \mathbb{D}} \quad \frac{\Gamma, x : \mathbb{C} \vdash f(x) : \mathbb{D}}{\Gamma, x : \mathbb{C} \vdash (\lambda x.f(x)) \cdot x = f(x) : \mathbb{D}}$$

$$\frac{\Gamma \vdash p : \mathbb{C} \times \mathbb{D}}{\Gamma \vdash \langle \pi_1(p), \pi_2(p) \rangle = p : \mathbb{C} \times \mathbb{D}} \quad \frac{\Gamma \vdash t : \top}{\Gamma \vdash t =\, ! : \top} \quad \frac{\Gamma \vdash s : \mathbb{C} \qquad \Gamma \vdash t : \mathbb{D}}{\Gamma \vdash \pi_1(\langle s, t \rangle) = s : \mathbb{C}} \quad \frac{\Gamma \vdash s : \mathbb{C} \qquad \Gamma \vdash t : \mathbb{D}}{\Gamma \vdash \pi_2(\langle s, t \rangle) = t : \mathbb{D}}$$

$$\frac{\Gamma \vdash t : \mathbb{C}}{\Gamma \vdash (t^{\mathsf{op}})^{\mathsf{op}} = t : \mathbb{D}} \quad \frac{\Gamma \vdash s : \mathbb{C} \qquad \Gamma \vdash t : \mathbb{D}}{\Gamma^{\mathsf{op}} \vdash \langle s, t \rangle^{\mathsf{op}} = \langle s^{\mathsf{op}}, t^{\mathsf{op}} \rangle : \mathbb{C}^{\mathsf{op}} \times \mathbb{D}^{\mathsf{op}}} \quad \frac{\Gamma^{\mathsf{op}}, x : \mathbb{C} \vdash t : \mathbb{D}}{\Gamma \vdash (\lambda x.t(x))^{\mathsf{op}} = \lambda x.t^{\mathsf{op}}(x) : [\mathbb{C}^{\mathsf{op}}, \mathbb{D}^{\mathsf{op}}]}$$

Fig. 8. Syntax of first-order dinatural directed type theory – contexts, variables, terms and their equality.

$$\boxed{[\Gamma]\ P\ \mathsf{prop}} \quad \frac{[\Gamma]\ P\ \mathsf{prop} \qquad [\Gamma]\ Q\ \mathsf{prop}}{[\Gamma]\ P \times Q\ \mathsf{prop}} \quad \frac{[\Gamma^{\mathsf{op}}]\ P\ \mathsf{prop} \qquad [\Gamma]\ Q\ \mathsf{prop}}{[\Gamma]\ P \Rightarrow Q\ \mathsf{prop}} \quad \frac{}{[\Gamma]\ \top\ \mathsf{prop}}$$

$$\frac{\Gamma^{\mathsf{op}}, \Gamma \vdash s : \mathbb{C}^{\mathsf{op}} \qquad \Gamma^{\mathsf{op}}, \Gamma \vdash t : \mathbb{C}}{[\Gamma]\ \hom_{\mathbb{C}}(s, t)\ \mathsf{prop}} \quad \frac{P \in \Sigma_P \qquad \Gamma^{\mathsf{op}}, \Gamma \vdash s : \mathsf{neg}(P)^{\mathsf{op}} \qquad \Gamma^{\mathsf{op}}, \Gamma \vdash t : \mathsf{pos}(P)}{[\Gamma]\ P(s \mid t)\ \mathsf{prop}}$$

$$\frac{[\Gamma, x : \mathbb{C}]\ P(\overline{x}, x)\ \mathsf{prop}}{[\Gamma]\ \int_{x:\mathbb{C}} P(\overline{x}, x)\ \mathsf{prop}} \quad \frac{[\Gamma, x : \mathbb{C}]\ P(\overline{x}, x)\ \mathsf{prop}}{[\Gamma]\ \int^{x:\mathbb{C}} P(\overline{x}, x)\ \mathsf{prop}}$$

$$\boxed{\Phi\ \mathsf{propctx}} \quad \frac{}{\bullet\ \mathsf{propctx}} \quad \frac{\Phi\ \mathsf{propctx} \qquad P\ \mathsf{prop}}{P, \Phi\ \mathsf{propctx}}$$

Fig. 9. Syntax of first-order dinatural directed type theory – predicates and propositional contexts.

$$\boxed{\Gamma \ni x : \mathbb{A} \ \text{cov in}\ P}$$

$$\frac{\Gamma \ni x : \mathbb{A}\ \text{cov in}\ P \qquad \Gamma \ni x : \mathbb{A}\ \text{cov in}\ Q}{\Gamma \ni x : \mathbb{A}\ \text{cov in}\ P \times Q} \qquad \frac{\Gamma^{\text{op}} \ni x : \mathbb{A}^{\text{op}}\ \text{cov in}\ P \qquad \Gamma \ni x : \mathbb{A}\ \text{cov in}\ Q}{\Gamma \ni x : \mathbb{A}\ \text{cov in}\ P \Rightarrow Q}$$

$$\frac{\Gamma^{\text{op}}, \Gamma \ni \overline{x} : \mathbb{A}^{\text{op}}\ \text{unused in}\ s : \mathbb{C}^{\text{op}} \qquad \Gamma^{\text{op}}, \Gamma \ni \overline{x} : \mathbb{A}^{\text{op}}\ \text{unused in}\ t : \mathbb{C}}{\Gamma \ni x : \mathbb{A}\ \text{cov in}\ \hom_{\mathbb{C}}(s,t)}$$

$$\frac{\Gamma^{\text{op}}, \Gamma \ni \overline{x} : \mathbb{A}^{\text{op}}\ \text{unused in}\ s : \text{neg}(P)^{\text{op}} \qquad \Gamma^{\text{op}}, \Gamma \ni \overline{x} : \mathbb{A}^{\text{op}}\ \text{unused in}\ t : \text{pos}(P)}{\Gamma \ni x : \mathbb{A}\ \text{cov in}\ P(s \mid t)}$$

$$\boxed{\Gamma \ni x : \mathbb{A}\ \text{contra in}\ P} \qquad \frac{\Gamma^{\text{op}} \ni x : \mathbb{A}^{\text{op}}\ \text{cov in}\ P^{\text{op}}}{\Gamma \ni x : \mathbb{A}\ \text{contra in}\ P}$$

$$\boxed{\Gamma \ni x : \mathbb{A}\ \text{unused in}\ t : \mathbb{C}} \qquad \frac{\Gamma \ni x : \mathbb{C} \qquad x \neq y}{\Gamma \ni y : \mathbb{C}\ \text{unused in}\ x : \mathbb{C}}$$

$$\frac{\Gamma \ni x : \mathbb{A}\ \text{unused in}\ t : \text{dom}(f)}{\Gamma \ni x : \mathbb{A}\ \text{unused in}\ f(t) : \text{cod}(f)} \qquad \frac{\Gamma \ni x : \mathbb{A}\ \text{unused in}\ t : \mathbb{C}}{\Gamma^{\text{op}} \ni x : \mathbb{A}^{\text{op}}\ \text{unused in}\ t^{\text{op}} : \mathbb{C}^{\text{op}}}$$

Fig. 10. Syntax of first-order dinatural directed type theory – syntactic conditions for covariant/contravariant variables in predicates. Full rules in Figure 14.

$$\boxed{[\Gamma]\ \Phi \vdash \alpha : P} \qquad \frac{}{[\Gamma]\ \Phi, a : P, \Phi' \vdash a : P}\ (\text{var}) \qquad \frac{[\Gamma]\ \Phi \vdash \alpha : Q}{[\Gamma]\ P, \Phi \vdash \text{wk}_P(\alpha) : Q}\ (\text{wk}) \qquad \frac{}{[\Gamma]\ \Phi \vdash ! : \top}\ (\top)$$

$$\frac{\Gamma^{\text{op}}, \Gamma \vdash F : \mathbb{C} \qquad [x : \mathbb{C}, \Gamma]\ \Phi(\overline{x}, x) \vdash \alpha : Q(\overline{x}, x)}{[\Gamma]\ \Phi(F(x,\overline{x}), F(\overline{x}, x)) \vdash F^*(\alpha) : Q(F(x,\overline{x}), F(\overline{x}, x))}\ (\text{idx}) \qquad \frac{[\Gamma]\ P, P, \Phi \vdash \alpha : Q}{[\Gamma]\ P, \Phi \vdash \text{contr}_P(\alpha) : Q}\ (\text{contr})$$

$$\frac{[\Gamma]\ \Phi \vdash P \times Q}{[\Gamma]\ \Phi \vdash P, \qquad [\Gamma]\ \Phi \vdash Q}\ (\text{prod}) \qquad \frac{[x : \Gamma]\ A(\overline{x}, x), \Phi(\overline{x}, x) \vdash B(\overline{x}, x)}{[x : \Gamma] \qquad \Phi(\overline{x}, x) \vdash A^{\text{op}}(x, \overline{x}) \Rightarrow B(\overline{x}, x)}\ (\text{exp})$$

$$\frac{[a : \mathbb{C}, \Gamma]\ \Phi \vdash P(\overline{a}, a)}{[\Gamma]\ \Phi \vdash \int_{a:\mathbb{C}} P(\overline{a}, a)}\ (\text{end}) \qquad \frac{[\Gamma]\ \left(\int^{a:\mathbb{C}} P(\overline{a}, a)\right), \Phi \vdash Q}{[a : \mathbb{C}, \Gamma]\ P(\overline{a}, a), \Phi \vdash Q}\ (\text{coend})$$

$$\frac{\begin{array}{c}\Gamma\ \text{unused in}\ P \\ [a{:}\Delta^{\text{op}}, b{:}\Delta]\ \Phi(a,b) \vdash \alpha \quad : P(a,b) \\ [z{:}\Delta]\ k{:}P(\overline{z}, z), \Phi(\overline{z}, z) \vdash \gamma[k] : Q(\overline{z}, z)\end{array}}{[z{:}\Delta]\ \Phi(\overline{z}, z) \vdash \gamma[\alpha] : Q(\overline{z}, z)}\ (\text{cut-din}) \qquad \frac{\begin{array}{c}\Gamma\ \text{unused in}\ P \\ [z{:}\Delta]\ \Phi(\overline{z}, z) \vdash \gamma \quad : P(\overline{z}, z) \\ [a{:}\Delta^{\text{op}}, b{:}\Delta]\ k{:}P(a,b), \Phi(\overline{a}, \overline{b}) \vdash \alpha[k] : Q(a,b)\end{array}}{[z{:}\Delta]\ \Phi(\overline{z}, z) \vdash \alpha[\gamma] : Q(\overline{z}, z)}\ (\text{cut-nat})$$

$$\frac{}{[x : \mathbb{C}, \Gamma]\ \Phi \vdash \text{refl}_{\mathbb{C}} : \hom_{\mathbb{C}}(\overline{x}, x)}\ (\text{refl}) \qquad \frac{[z : \mathbb{C}, \Gamma] \qquad \Phi(\overline{z}, z) \ \vdash h : P(\overline{z}, z)}{[a : \mathbb{C}^{\text{op}}, b : \mathbb{C}, \Gamma]\ e : \hom_{\mathbb{C}}(a,b), \Phi(\overline{b}, \overline{a}) \vdash J(h)[e] : P(a,b)}\ (J)$$

$$\boxed{[\Gamma]\ \Phi \vdash \alpha = \beta : P} \qquad \frac{}{[z : \mathbb{C}, \Gamma]\ k : \Phi(\overline{z}, z) \vdash J(h)[\text{refl}_{\mathbb{C}}] = h : P(\overline{z}, z)}\ (J\text{-comp})$$

$$\frac{[z : \mathbb{C}, \Gamma]\ \Phi(\overline{z}, z) \vdash \alpha[\text{refl}_{\mathbb{C}}] = \beta[\text{refl}_{\mathbb{C}}] : P(\overline{z}, z)}{[a : \mathbb{C}^{\text{op}}, b : \mathbb{C}, \Gamma]\ e : \hom_{\mathbb{C}}(a,b), \Phi(\overline{b}, \overline{a}) \vdash \alpha[e] = \beta[e] : P(a,b)}\ (J\text{-eq})$$

Fig. 11. Syntax of first-order dinatural directed type theory – entailments and judgmental equality.

The rules for entailments implicitly use the notion of variance for variables, described in Remark 2.2. Variance is captured formally in Figures 10 and 13 by the following judgments, all of which presuppose $\Gamma \ni x : \mathbb{A}$ for a variable x of type $\mathbb{A}$ in context Γ:

Figure 13:
- $\boxed{\Gamma \ni x : \mathbb{A} \text{ unused in } t : \mathbb{C}}$ for $x{:}\mathbb{A}$ does not syntactically appear in a term t.

Figure 10:
- $\boxed{\Gamma \ni x : \mathbb{A} \text{ cov in } P}$ states that $x{:}\mathbb{A}$ is *covariant* in the predicate $[\Gamma]\ P$.
- $\boxed{\Gamma \ni x : \mathbb{A} \text{ contra in } P}$ states that $x{:}\mathbb{A}$ is *contravariant* in the predicate $[\Gamma]\ P$.

To make the type theory non-trivial, our judgments are implicitly parameterized by a standard notion of signature $\Sigma := (\Sigma_B, \Sigma_T, \Sigma_P, \Sigma_A)$, i.e., sets of base type symbols, term symbols, predicate symbols, and base entailments respectively. Base predicates $P(s \mid t)$ for $P \in \Sigma_P$ are equipped with *two* terms, a negative one $s : \mathrm{neg}(P)^{\mathrm{op}}$ and a positive one $t : \mathrm{pos}(P)$ typed in the same context $\Gamma^{\mathrm{op}}, \Gamma$. This choice is motivated by the fact that hom is similarly equipped with two sides. The judgments for equality of types are not extended by the signature. We omit the details of this extension.

2.1 Polarity and variance

The main idea behind dinatural transformations is that variables in a predicate are allowed to be used irrespectively of the op in their type (or lack thereof). To give a taste for our type theory, we show what the statement and proof of transitivity of directed equality look like in our system:

$$\dfrac{\dfrac{}{[z : \mathbb{C}, c : \mathbb{C}] \qquad g : \hom(\overline{z}, c) \vdash g : \hom(\overline{z}, c)}\ (\mathrm{var})}{[a : \mathbb{C}^{\mathrm{op}}, b : \mathbb{C}, c : \mathbb{C}]\ f : \hom(a, b),\ g : \hom(\overline{b}, c) \vdash J(g) : \hom(a, c)}\ (J)$$

Whenever a variable $b : \mathbb{C}$ is used with the "wrong polarity" we denote such use with $\overline{b} : \mathbb{C}^{\mathrm{op}}$, as in the above example. In order to make this intuition precise, we formally introduce the concepts of *position*, *polarity*, and *variance* and their notation in the type theory. Variance is ultimately used to implement the syntactic restriction of directed equality elimination (J).

We use the term *polarity of a type* to refer to the fact that types always come in pairs: whenever $\mathbb{C}$ is a type, its opposite $\mathbb{C}^{\mathrm{op}}$ is also a type. Polarity is a relative notion: we say the type $\mathbb{C}^{\mathrm{op}}$ is *the negative* of $\mathbb{C}$ irrespectively of the fact that $\mathbb{C}$ itself might have an outermost syntactic op. Polarity is used in the syntax of the type theory in the following way:

- The op operation is also present in contexts, i.e., for a Γ ctx there is a *negative context* Γ^{op} which is definitionally equal to the context obtained by adding op to each type.
- In the formation rule for $[\Gamma]\ \hom_{\mathbb{C}}(s, t)$ in Figure 9, the term s is given return type $\mathbb{C}^{\mathrm{op}}$.
- In the formation rule for $[\Gamma]\ P \Rightarrow Q$ in Figure 9, the predicate P is given type in Γ^{op}.

The other crucial idea of our system is the above-mentioned fact that variables can appear at the same time irrespectively of their polarity. This is implemented by the following ideas:

- There are two cases where variables can appear in a predicate, namely the base cases $[\Gamma]\ \hom_{\mathbb{C}}(s, t)$ and $[\Gamma]\ P(s \mid t)$, where the two terms s, t can use the variables from Γ.
- The key idea is that both s, t are not given type in Γ, but in the *context concatenation* $\Gamma^{\mathrm{op}}, \Gamma$.
- Intuitively, this allows for variables to be used in s, t also in the "wrong way" (with respect to the original polarity of the context Γ in which P is given type).

We give a specific name to the terms of this shape in concatenated contexts $\Gamma^{\mathrm{op}}, \Gamma$, since they also play a crucial role in reindexing.

Definition 2.1. A *diterm* is a term of the form $\Gamma^{\mathrm{op}}, \Gamma \vdash t : \mathbb{C}$ for some context Γ.

We now capture the above intuitive ideas behind polarity and variance with precise terminology.

Definition 2.2 (Positions in a predicate). The name *position* refers to a point in which a variable $x : \mathbb{C}$ can appear in a predicate, e.g., there are four possible positions x, y, z, w for variables to appear in the predicate $\hom_{\mathbb{C}}(x, y) \times P(z, F(w))$.

Definition 2.3 (Variant use of a variable). For any predicate $[\Gamma]\ P$ and a position of type $\mathbb{C}^{\text{op}}$ in P, we say that a variable $\Gamma \ni x : \mathbb{C}$ (with no op) is *used contravariantly in that position* iff the variable used in that position is taken from the *left* side Γ^{op} (in the context concatenation $\Gamma^{\text{op}}, \Gamma$), i.e., with type $\overline{x} : \mathbb{C}^{\text{op}}$. Accordingly, we will always denote variables taken from such left side of the context with an overbar $\overline{x}$. Similarly, given a position of type $\mathbb{C}$ in P we say that a variable $\Gamma \ni x : \mathbb{C}$ is *used covariantly in that position* iff it is taken from the *right* side Γ (i.e. in the usual way), which we denote without any overbar.

The notation $\overline{x}$ is suggestive of the fact that $\overline{x} : \mathbb{C}^{\text{op}}$ and $x : \mathbb{C}$ will be identified with the same value when using dinatural transformations in the semantics of entailments.

Example 2.4 (Derivation of a predicate). We provide an example derivation of a predicate in context combining the previously introduced ideas of co/contravariant variables, for a term $x : \mathbb{C} \vdash F(x) : \mathbb{D}$.

$$
\dfrac{\dfrac{\dfrac{}{\overline{x}:\mathbb{C}, \overline{y}:\mathbb{D}, x:\mathbb{C}^{\text{op}}, y:\mathbb{D}^{\text{op}} \vdash y:\mathbb{D}^{\text{op}}} \qquad \dfrac{\dfrac{}{\overline{x}:\mathbb{C}, \overline{y}:\mathbb{D}, x:\mathbb{C}^{\text{op}}, y:\mathbb{D}^{\text{op}} \vdash \overline{x} : \mathbb{C}}}{\overline{x}:\mathbb{C}, \overline{y}:\mathbb{D}, x:\mathbb{C}^{\text{op}}, y:\mathbb{D}^{\text{op}} \vdash F(\overline{x}) : \mathbb{D}}}{[x:\mathbb{C}^{\text{op}}, y:\mathbb{D}^{\text{op}}]\ \hom_{\mathbb{D}}(y, F(\overline{x}))\ \mathsf{prop}} \qquad \dfrac{\cdots \vdash x : \mathbb{C}}{[x:\mathbb{C}, y:\mathbb{D}]\ P(x)\ \mathsf{prop}}}{[x:\mathbb{C}, y:\mathbb{D}]\ \hom_{\mathbb{D}}(y, F(\overline{x})) \Rightarrow P(x)\ \mathsf{prop}}
$$

Definition 2.5 (Variance of a variable). Variables can occur in multiple positions at the same time: we say that a variable $\Gamma \ni x : \mathbb{C}$ is *covariant* in a predicate $[\Gamma]\ P$ *iff* it is *always used covariantly* in the positions of P, i.e., it is always picked from the right side Γ of the context $\Gamma^{\text{op}}, \Gamma$ and is hence always used "correctly" with respect to Γ. Similarly, a variable $\Gamma \ni x : \mathbb{C}$ is said to be *contravariant* in a predicate $[\Gamma]\ P$ when it is *always used contravariantly* in the positions of P, i.e., it is always picked from the left side Γ^{op} of the context $\Gamma^{\text{op}}, \Gamma$ and is hence always used "in the wrong way" with respect to Γ. A variable is said to be *natural* when it is either covariant or contravariant, i.e., it is consistently used with the same variance. A variable is said to be *dinatural* or *mixed-variance iff* it is neither covariant nor contravariant, i.e., it occurs at least once covariantly and at least once contravariantly in a predicate.

Example 2.6 (Variance). In the predicate $[x:\mathbb{C}^{\text{op}}, y:\mathbb{C}]\ \hom_{\mathbb{C}}(x, y)$, both x and y are covariant. In $[x:\mathbb{C}, y:\mathbb{C}, z:\mathbb{C}]\ \hom_{\mathbb{C}}(\overline{x}, y) \times \hom_{\mathbb{C}}(\overline{y}, z)$ the variable x is contravariant, y is dinatural, and z is covariant. In $[x:\mathbb{C}^{\text{op}}, z:\mathbb{C}^{\text{op}}]\ \hom_{\mathbb{C}}(\overline{x}, z) \Rightarrow \hom_{\mathbb{C}}(z, \overline{x})$, x is contravariant and z is covariant. Finally, for a term $\mathbb{C}^{\text{op}} \vdash F : \mathbb{D}$ (i.e., a "contravariant functor"), x is covariant in $[x:\mathbb{C}]\ \hom_{\mathbb{D}}(F(x), x)$.

The above definitions capture the way that natural and dinatural usage of variables is referred to in practice. Formally, variance of variables in predicates is captured using the judgments in Figures 10 and 13. The actual implementation of variance is slightly different from the description above, but they are equivalent: the judgment $\Gamma \ni x : \mathbb{A}$ cov in P is derivable, i.e., the variable x is covariant, when *its contravariant counterpart* $\overline{x}$ *is not syntactically used anywhere in the predicate.* This last aspect is itself captured by a straightforward judgment, described in Figure 10, which underapproximates syntactic unusedness of variables in terms. The well-formedness of these judgments occasionally relies on the fact that $\Gamma \ni x : \mathbb{A}$ implies that $\Gamma^{\text{op}} \ni x : \mathbb{A}^{\text{op}}$, and similarly $\Gamma^{\text{op}}, \Gamma \ni x : \mathbb{A}$ and $\Gamma^{\text{op}}, \Gamma \ni \overline{x} : \mathbb{A}^{\text{op}}$ in the intuitive way.

Example 2.7 (Variance, formally). We give an example of a formal derivation for covariance using the predicate in Figure 10, assuming for simplicity that the predicate P does not have any variables:

$$\dfrac{\dfrac{\dfrac{}{[\overline{x}:\mathbb{C},\overline{y}:\mathbb{D},x:\mathbb{C}^{\mathrm{op}},y:\mathbb{D}^{\mathrm{op}}] \ni \overline{y}:\mathbb{D} \text{ unused in } y} \qquad \dfrac{}{[\cdots] \ni \overline{y}:\mathbb{D} \text{ unused in } F(x)}}{[x:\mathbb{C}^{\mathrm{op}},y:\mathbb{D}^{\mathrm{op}}] \ni y:\mathbb{D}^{\mathrm{op}} \text{ cov in } \hom_{\mathbb{D}}(y,F(\overline{x}))} \qquad \dfrac{}{\cdots}}{[x:\mathbb{C},y:\mathbb{D}] \ni y:\mathbb{D} \text{ cov in } \hom_{\mathbb{D}}(y,F(\overline{x})) \Rightarrow P}$$

Remark (Notation for variance in predicates). *We indicate with* $[x:\mathbb{C},y,\mathbb{D},\Gamma]\ P(\overline{x},x,\overline{y},y)$ *the fact that a predicate* P can *depend on* x, y *both co- and contravariantly; we will often omit in* P *the (unrestricted) presence of variables coming from a context* Γ*. When either variance is omitted, e.g., as in* $P(x,\overline{y})$*, the predicate must depend* only *on* x *and* $\overline{y}$*, i.e.,* x *is* covariant *and* $\overline{y}$ *is* contravariant *in* P*. Variance for entire contexts is intuitively denoted as* $[y:\Gamma]\ P(y)$*, i.e., all variables in* Γ *are covariant.*

Formally, these restrictions are captured using the predicates for variance of Definition 2.5. We use this convention in the rules for entailments of Figure 11.

There are many choices for the system of variances presented so far: the one presented here is a simple setup that closely matches the intuition for contravariance typically used in mathematics, denoting variables as contravariant precisely when one expects it as shown in Example 2.4.

Mnemonically, *p*ositions have *p*olarity, and *v*ariables have *v*ariance. *Co*variant variables are "*co*mpliant" and they are used as they are told, while *contra*variant variables are "*contra*rian" and always reject well-typing laws.

For any predicate $[\Gamma]\ P$, there is an associated *opposite predicate* $[\Gamma^{\mathrm{op}}]\ P^{\mathrm{op}}$, defined by induction on the derivation of P, obtained intuitively by inverting the variance of variables in each position: i.e., whenever x was used in some position, $\overline{x}$ is used instead, and vice versa. This operation is used in the rule for polarized implication (exp), described in Section 2.2, and to define contravariance in Figure 10. Note that this operation on predicates is defined metatheoretically: types and terms are the only two judgments for which there is a $-^{\mathrm{op}}$ *in the syntax.*

We start by first defining a metatheoretical operation on diterms that simply swaps contexts:

Definition 2.8 (Context swap of a term). Given a diterm $\Gamma^{\mathrm{op}},\Gamma \vdash t:\mathbb{C}$, we indicate with $\Gamma,\Gamma^{\mathrm{op}} \vdash t^{\mathrm{ctxswap}} : \mathbb{C}$ the *context swap* of t, which is the term derivation obtained in the intuitive way by swapping the left and right side of its context; for example, $(\overline{x}:\mathbb{D}^{\mathrm{op}},x:\mathbb{D} \vdash x:\mathbb{D})^{\mathrm{ctxswap}} = (\overline{x}:\mathbb{D},x:\mathbb{D}^{\mathrm{op}} \vdash \overline{x}:\mathbb{D})$, and $(\overline{x}:\mathbb{C}^{\mathrm{op}},x:\mathbb{C} \vdash F(\overline{x}):\mathbb{D})^{\mathrm{ctxswap}} = (\overline{x}:\mathbb{C},x:\mathbb{C}^{\mathrm{op}} \vdash F(x):\mathbb{D})$ for some term $\overline{x}:\mathbb{C}^{\mathrm{op}},x:\mathbb{C} \vdash F(x):\mathbb{D}$. Crucially, the return type of the term does not change, which would be the case with the t^{op} operation internal to the syntax. Effectively this operation only rearranges the de Bruijn indices of variables, which is what the judgments for variance in Figure 10 use to detect co/contravariance.

Definition 2.9 (Opposite predicate). Given a predicate $[\Gamma]\ P$, there is a predicate in context Γ^{op} called *the opposite of* P defined by (metatheoretical) induction on derivations of predicates:

$$\begin{aligned}
&-^{\mathrm{op}} : \{[\Gamma] - \mathrm{prop}\} \longrightarrow \{[\Gamma^{\mathrm{op}}] - \mathrm{prop}\}\\
&(\top)^{\mathrm{op}} := \top\\
&(P \Rightarrow Q)^{\mathrm{op}} := P^{\mathrm{op}} \Rightarrow Q^{\mathrm{op}}\\
&(P \times Q)^{\mathrm{op}} := P^{\mathrm{op}} \times Q^{\mathrm{op}}\\
&(P(s \mid t))^{\mathrm{op}} := P(s^{\mathrm{ctxswap}} \mid t^{\mathrm{ctxswap}})\\
&(\hom_{\mathbb{C}}(s,t))^{\mathrm{op}} := \hom_{\mathbb{C}}(s^{\mathrm{ctxswap}}, t^{\mathrm{ctxswap}})\\
&\left(\int^{x:\mathbb{C}} P(\overline{x},x)\right)^{\mathrm{op}} := \int^{x:\mathbb{C}^{\mathrm{op}}} P(\overline{x},x)^{\mathrm{op}}\\
&\left(\int_{x:\mathbb{C}} P(\overline{x},x)\right)^{\mathrm{op}} := \int_{x:\mathbb{C}^{\mathrm{op}}} P(\overline{x},x)^{\mathrm{op}}
\end{aligned}$$

This operation can similarly be defined by inverting the polarity of a single variable: given a predicate $[x : \mathbb{C}, \Gamma]\ P(\overline{x}, x)$ we denote with $[x : \mathbb{C}^{\mathrm{op}}, \Gamma]\ P^{x \mapsto \mathrm{op}}(x, \overline{x})$ the predicate obtained by inverting the polarity of each position in P where x is used. A similar definition can be extended on propositional contexts Φ. All these operations on predicates are clearly involutive.

Example 2.10. Taking the predicate of Example 2.4 and applying the predicate inversion operation $(\hom_{\mathbb{D}}(y, F(\overline{x})))^{\mathrm{op}}$ produces the following derivation:

$$\dfrac{\dfrac{}{\overline{x}:\mathbb{C}^{\mathrm{op}}, \overline{y}:\mathbb{D}^{\mathrm{op}}, x:\mathbb{C}, y:\mathbb{D} \vdash \overline{y}:\mathbb{D}^{\mathrm{op}}} \qquad \dfrac{\dfrac{}{\overline{x}:\mathbb{C}^{\mathrm{op}}, \overline{y}:\mathbb{D}^{\mathrm{op}}, x:\mathbb{C}, y:\mathbb{D} \vdash x : \mathbb{C}}}{\overline{x}:\mathbb{C}^{\mathrm{op}}, \overline{y}:\mathbb{D}^{\mathrm{op}}, x:\mathbb{C}, y:\mathbb{D} \vdash F(x) : \mathbb{D}}}{[x:\mathbb{C}, y:\mathbb{D}]\ \hom_{\mathbb{D}}(\overline{y}, F(x))\ \mathsf{prop}}$$

The judgment for contravariance $\Gamma \ni x : \mathbb{A}$ contra in P in Figure 10 is defined in terms of the covariant one and the notion of opposite predicate P^{op}. Note that the well-formedness of this judgment relies on the fact that $\Gamma \ni x : \mathbb{C}$ implies $\Gamma^{\mathrm{op}} \ni x : \mathbb{C}^{\mathrm{op}}$.

Example 2.11 (Contravariance, formally). We give an example of a formal derivation for contravariance, following Example 2.7:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{[\cdots] \ni \overline{x} : \mathbb{C}^{\mathrm{op}} \text{ unused in } \overline{y}} \qquad \dfrac{\dfrac{}{[\overline{x}:\mathbb{C}^{\mathrm{op}}, \overline{y}:\mathbb{D}^{\mathrm{op}}, x:\mathbb{C}, y:\mathbb{D}] \ni \overline{x} : \mathbb{C}^{\mathrm{op}} \text{ unused in } x : \mathbb{D}}}{[\overline{x}:\mathbb{C}^{\mathrm{op}}, \overline{y}:\mathbb{D}^{\mathrm{op}}, x:\mathbb{C}, y:\mathbb{D}] \ni \overline{x} : \mathbb{C}^{\mathrm{op}} \text{ unused in } F(x) : \mathbb{D}}}{[x:\mathbb{C}, y:\mathbb{D}] \ni x : \mathbb{C} \text{ cov in } \hom_{\mathbb{D}}(\overline{y}, F(x))} \qquad \dfrac{}{\cdots}}{[x:\mathbb{C}^{\mathrm{op}}, y:\mathbb{D}^{\mathrm{op}}] \ni x : \mathbb{C}^{\mathrm{op}} \text{ cov in } \hom_{\mathbb{D}}(\overline{y}, F(x)) \Rightarrow P}}{[x:\mathbb{C}, y:\mathbb{D}] \ni x : \mathbb{C} \text{ contra in } \hom_{\mathbb{D}}(y, F(\overline{x})) \Rightarrow P}$$

2.2 Rules

We now describe and give intuition for the main rules for entailments of our type theory in Figure 10.

Remark (Notation for entailments). *We use type-theoretic notation for entailments,*

$$[x : \mathbb{C}, y : \mathbb{D}, ...]\ a : P(\overline{x}, x, \overline{y}, y, ...), b : Q(\overline{x}, x, \overline{y}, y, ...), ... \vdash \alpha[a, b, ...] : R(\overline{x}, x, \overline{y}, y, ...)$$

where we give names to each assumption in the list $\Phi := P, Q, ...$ We overload square brackets $\alpha[a, b, ...]$ both to indicate the assumptions and to denote composition of entailments in (cut-din) and (cut-nat).

Some of our rules are formulated in "adjoint-form" (e.g. [43, 4.1.7, 4.1.8]), i.e., as natural *bijections* between entailments. We use double lines in Figure 11 to indicate such isomorphisms of entailments, using judgmental equality of entailments to ensure that one direction is the inverse of the other. Naturality coincides with the fact that these isomorphisms commute with (both) the cut rules in the equational theory whenever possible: we use this in Section 6 for the Yoneda technique. We give a spelled-out example of adjoint-form in Figure 16 for the (end) rule, describing precisely the naturality requirement for the rules in such form.

- **Structural rules.** The rules (var), (wk), (contr) capture the usual structural rules for assumptions, weakening, and contraction.
- **Products.** The rule (prod) for conjunction $P \times Q$ is standard: reading the rule top-to-bottom, given a proof $[\Gamma]\ \Phi \vdash P \times Q$ one can extract a proof $[\Gamma]\ \Phi \vdash P$. Similarly, given two entailments with type P and Q in the same context one obtains an entailment with type $P \times Q$.
- **Polarized implication.** Implication (exp) is similarly captured via the adjoint formulation, with a catch regarding polarity: the key idea is that a predicate $P(\overline{x}, x)$ can be curried from one side to the other of the entailment *by reversing the variance of all its variables*, i.e., using P^{op}. Contrary

to naturals and presheaves [50], dinaturals can be curried directly via the (exp) rule by currying each component of α in Set. A similar idea is described in [10, 32] as *twisted exponential*.

- **(Co)ends.** The rules (end), (coend) capture the directed quantifiers of our type theory, i.e., (co)ends. These are also characterized in "adjoint-form", following precisely the same formulation of [43, 4.1.8]. Note that Φ is given type in Γ, and we do not make this weakening explicit.
- **Reindexing.** Following the doctrinal presentation of logic (see [43, 71] for standard accounts), variables in entailments can be substituted with terms using the rule (idx): in particular, entailments can be substituted with *diterms*, i.e., terms that are allowed to access the *whole concatenation of contexts* $\Gamma^{\mathrm{op}}, \Gamma$. The fact that F is a *di*term is not a mere technical point, and it is used in Remark 3.2 and theorem 3.14 to derive certain non-trivial structural rules related to variance.
- **Cut naturals-dinaturals.** We present two restricted cut rules (cut-din), (cut-nat) that allow entailments to be composed together. Associativity and identities for these is captured in Figure 15, along with a coherence condition that makes the two cuts agree whenever both entailments are *naturals*. The occurrences $\overline{a}, \overline{b}$ in Φ in (cut-nat) are needed to make sure that, in the semantics, α is natural in a, b when the domain is *just* P, i.e., by using (exp) to move Φ and invert the variance of $\overline{a}, \overline{b}$. Similarly, P must also not syntactically depend on Γ to ensure naturality in a, b, but both Φ and Q can depend on Γ without any restriction; we elaborate on this in the semantics of cuts in Section 5, which we use to state the naturality requirement for, e.g., ends in Figure 16.
- **Directed equality elimination.** The operational meaning behind (J) is the following: having identified two *covariant* positions $a : \mathbb{C}^{\mathrm{op}}$ and $b : \mathbb{C}$ in the predicate P, if there is a directed equality $\hom_{\mathbb{C}}(a, b)$ in context then it is enough to prove that P holds "on the diagonal", where the two positions have been collapsed with the same dinatural variable $z : \mathbb{C}$; moreover, a, b can be collapsed together in the context Φ *only if they appear contravariantly*, i.e., as $\overline{a}$ and $\overline{b}$.
- **Dependent** hom **elimination.** A *dependent* version of directed J, rule (J-eq), is needed to prove equational properties of maps definable with (J); this is done by allowing $\hom(a, b)$ to be contracted *inside equality judgments*. Intuitively, given entailments $\alpha[e]$ and $\beta[e]$ with an equality in context $e : \hom_{\mathbb{C}}(a, b)$ which can be contracted using (J), we can deduce that α and β are equal everywhere as soon as they are equal on $e = \mathsf{refl}_{\mathbb{C},z}$ for every $z : \mathbb{C}$.

3 Directed equality à la Martin-Löf

We show how the rules for directed equality can be used to obtain the same terms definable with symmetric equality in Martin-Löf type theory, and proving properties about them follows precisely the steps of the usual proofs, i.e., by equality contraction and computation rules [41, 78]. All examples in this section satisfy the constraints for (cut-nat), (cut-din) to be applied.

We start by showing transitivity of directed equality, i.e., categories have composition maps.

Example 3.1 (Composition in a category). The following derivation constructs the *composition map for* $\mathbb{C}$, which is covariant in $a : \mathbb{C}^{\mathrm{op}}, c : \mathbb{C}$ and dinatural in $b : \mathbb{C}$:

$$\dfrac{\dfrac{}{[z : \mathbb{C}, c : \mathbb{C}] \quad g : \hom(\overline{z}, c) \vdash g : \hom(\overline{z}, c)}\,(\mathrm{var})}{[a : \mathbb{C}^{\mathrm{op}}, b : \mathbb{C}, c : \mathbb{C}]\ f : \hom(a, b),\ g : \hom(\overline{b}, c) \vdash J(g) : \hom(a, c)}\,(J)$$

We contracted the first equality $f : \hom(a, b)$. Rule (J) can be applied since a, b appear only contravariantly in context (a does not appear) and covariantly in the conclusion ($\overline{b}$ does not). We now prove that $\mathrm{comp}[f, g] := J(g)$, denoted as "$f \,;\, g$", is unital on identities (i.e., $\mathsf{refl}_{\mathbb{C}}$) and associative. Since we chose to contract f, the computation rule ensures unitality on the left:

$$\dfrac{}{[z : \mathbb{C}, c : \mathbb{C}]\ g : \hom(\overline{z}, c) \vdash \mathsf{refl}_z \,;\, g = g : \hom(\overline{z}, c)}\,(J\text{-comp})$$

On the other hand, to show that composition is right-unital we use dependent directed equality induction (J-eq), where now it is enough to just consider the case in which $a = z = w$ and $f = \mathsf{refl}_w$,

$$\dfrac{\dfrac{}{[w : \mathbb{C}]\ \bullet \vdash \mathsf{refl}_w \mathbin{;} \mathsf{refl}_w = \mathsf{refl}_w : \hom(\overline{w}, w)}\ (J\text{-comp})}{[a : \mathbb{C}^{\mathsf{op}}, z : \mathbb{C}]\ f : \hom(a, z) \vdash f \mathbin{;} \mathsf{refl}_z = f : \hom(a, z)}\ (J\text{-eq})$$

which follows by the computation rule for comp since refl_w is on the left. Similarly, to show associativity we just need to consider the case $a = b = z$ and $f = \mathsf{refl}_z$,

$$\dfrac{\dfrac{}{[z : \mathbb{C}, c : \mathbb{C}, d : \mathbb{C}]\quad g : \hom(\overline{z}, c), h : \hom(\overline{c}, d) \ \vdash \mathsf{refl}_z \mathbin{;} (g \mathbin{;} h) = (\mathsf{refl}_z \mathbin{;} g) \mathbin{;} h : \hom(\overline{z}, d)}\ (J\text{-comp})}{[a : \mathbb{C}, b : \mathbb{C}, c : \mathbb{C}, d : \mathbb{C}]\ f : \hom(\overline{a}, b), g : \hom(\overline{b}, c), h : \hom(\overline{c}, d) \vdash f \mathbin{;} (g \mathbin{;} h) = (f \mathbin{;} g) \mathbin{;} h : \hom(\overline{a}, d)}\ (J\text{-eq})$$

where in the top sequent both entailments are equal to $g \mathbin{;} h$ by the computation rules of comp.

Example 3.2 (Functorial action on morphisms). For any term/functor $\mathbb{C} \vdash F : \mathbb{D}$, the functorial action on morphisms of F corresponds with the fact that any term F respects directed equality, i.e., directed equality is a congruence:

$$\dfrac{\dfrac{}{[z : \mathbb{C}]\quad \bullet \vdash F^*(\mathsf{refl}_{\mathbb{C}}) : \hom_{\mathbb{D}}(F^{\mathsf{op}}(\overline{z}), F(z))}\ (\mathsf{idx}){+}(\mathsf{refl})}{[x : \mathbb{C}, y : \mathbb{C}]\ f : \hom_{\mathbb{C}}(\overline{x}, y) \vdash J(F^*(\mathsf{refl}_{\mathbb{C}})) : \hom_{\mathbb{D}}(F^{\mathsf{op}}(\overline{x}), F(y))}\ (J)$$

and thus we define $\mathsf{map}_F[f] := J(F^*(\mathsf{refl}_{\mathbb{C}}))$, using (idx) with F in the top sequent.
The computation rule states that F maps identities to identities:

$$\dfrac{}{[z : \mathbb{C}]\ \top \vdash \mathsf{map}_F[\mathsf{refl}_{\mathbb{C}}] = F^*(\mathsf{refl}_{\mathbb{C}}) : \hom_{\mathbb{D}}(F^{\mathsf{op}}(\overline{x}), F(x))}\ (J\text{-comp})$$

The following shows functoriality for free; both top sides reduce to $\mathsf{map}_F[g]$ using (J-comp):

$$\dfrac{\dfrac{}{[z : \mathbb{C}, c : \mathbb{C}]\quad g : \hom(\overline{z}, c) \ \vdash \mathsf{map}_F[\mathsf{refl}_z \mathbin{;} g] = \mathsf{refl}_{F(z)} \ \mathbin{;} \mathsf{map}_F[g] : \hom(\overline{z}, d)}\ (J\text{-comp})}{[a : \mathbb{C}, b : \mathbb{C}, c : \mathbb{C}]\ f : \hom(\overline{a}, b), g : \hom(\overline{b}, c) \vdash \mathsf{map}_F[f \mathbin{;} g] = \mathsf{map}_F[f] \mathbin{;} \mathsf{map}_F[g] : \hom(\overline{a}, d)}\ (J\text{-eq})$$

Example 3.3 (Transport). Transporting points of predicates along directed equalities [78, 2.3.1] is the functorial action of copresheaves $P : \mathbb{C} \to \mathsf{Set}$, i.e., predicates $[x : \mathbb{C}]\ P$ prop, for x only positive:

$$\dfrac{\dfrac{}{[z : \mathbb{C}]\ k : P(z) \vdash k : P(z)}\ (\mathsf{var})}{[a : \mathbb{C}^{\mathsf{op}}, b : \mathbb{C}]\ f : \hom(a, b), k : P(\overline{a}) \vdash J(k) : P(b)}\ (J)$$

The computation rule simply states that transporting a point of $P(a)$ along the identity morphism with $\mathsf{subst}[f, k] := J(k)$ is the same as giving the point itself, i.e., $\mathsf{subst}[\mathsf{refl}_{\mathbb{C}}, k] = k$.

Example 3.4 (Pair of rewrites). Pairs of directed equalities induce directed equalities between pairs. The other direction (i.e., "directed injectivity of pairs") follows from congruence of directed equality with the projections π_1, π_2 and then using the judgmental equality of terms.

$$\dfrac{\dfrac{\dfrac{}{[z : \mathbb{C}, z' : \mathbb{D}]\ \bullet \vdash \hom_{\mathbb{C}\times\mathbb{D}}((\overline{z}, \overline{z}), (z, z))}\ (\mathsf{idx}){+}(\mathsf{refl})}{[a' : \mathbb{C}^{\mathsf{op}}, b' : \mathbb{D}, z : \mathbb{C}]\ g : \hom_{\mathbb{D}}(b, b') \vdash \hom_{\mathbb{C}\times\mathbb{D}}((\overline{z}, b), (z, b'))}\ (J)}{[a, a' : \mathbb{C}^{\mathsf{op}}, b, b' : \mathbb{D}]\ f : \hom_{\mathbb{C}}(a, a'), g : \hom_{\mathbb{D}}(b, b') \vdash \hom_{\mathbb{C}\times\mathbb{D}}((a, b), (a', b'))}\ (J)$$

Example 3.5 (Higher-dimensional rewriting). The following shows that a directed equality between functors induces a natural transformation [52, 1.4.1] (omitting the resulting term for simplicity):

$$\dfrac{\dfrac{\dfrac{}{[H:[\mathbb{C},\mathbb{D}],x:\mathbb{C}]\;\bullet\vdash \hom_{\mathbb{D}}(\overline{H}\cdot\overline{x},H\cdot x)}\;(\text{idx})+(\text{refl})}{[H:[\mathbb{C},\mathbb{D}]]\;\bullet\vdash \int_{x:\mathbb{C}}\hom_{\mathbb{D}}(\overline{H}\cdot\overline{x},H\cdot x)}\;(\text{end})}{[F:[\mathbb{C},\mathbb{D}]^{\text{op}},G:[\mathbb{C},\mathbb{D}]]\;e:\hom_{[\mathbb{C},\mathbb{D}]}(F,G)\vdash \int_{x:\mathbb{C}}\hom_{\mathbb{D}}(F\cdot\overline{x},G\cdot x)}\;(J)$$

The opposite direction is not derivable in general, since in the case where $\mathbb{C},\mathbb{D}$ are discrete categories (i.e., sets), it corresponds to function extensionality.

Example 3.6 (Existence of singletons). The following derivation asserts that singleton subsets are inhabited [78, Remark 1.12.1], i.e., there is a proof for the first-order logic formula $\forall x.\exists y.x=y$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{[x:\mathbb{C}^{\text{op}}]\;k:\int^{y:\mathbb{C}}\hom_{\mathbb{C}}(x,y)\vdash k:\int^{y:\mathbb{C}}\hom_{\mathbb{C}}(x,y)}\;(\text{var})}{[x:\mathbb{C}^{\text{op}},y:\mathbb{C}]\;f:\hom_{\mathbb{C}}(x,y)\vdash \text{coend}^{-1}(k)[f]:\int^{y:\mathbb{C}}\hom_{\mathbb{C}}(x,y)}\;(\text{coend})}{[x:\mathbb{C}]\;\bullet\vdash \text{coend}^{-1}(k)[\text{refl}_x]:\int^{y:\mathbb{C}}\hom_{\mathbb{C}}(\overline{x},y)}\;(\text{cut-nat})}{[\,]\;\bullet\vdash \text{end}(\text{coend}^{-1}(k)[\text{refl}_x]):\int_{x:\mathbb{C}}\int^{y:\mathbb{C}}\hom_{\mathbb{C}}(\overline{x},y)}\;(\text{end})$$

This derivation is actually an isomorphism in the model, i.e., singletons are contractible. This follows from dependent directed equality contraction, which we show in detail in Example B.1.

The following theorems show that in our type theory both naturality and dinaturality follow "for free" from dependent directed equality contraction. Cuts are allowed in both cases because of the *natural* appearance of variables in subst.

Example 3.7 (Internal naturality for entailments). For any $[x:\mathbb{C}]\;P(x)\vdash\alpha:Q(x)$, an internal version of naturality for entailments holds via (J-comp):

$$\dfrac{\dfrac{}{[z:\mathbb{C}]\;k:P(z)\vdash \alpha[\text{subst}_P[\text{refl}_z,k]]=\text{subst}_Q[\text{refl}_z,\alpha[k]]:Q(z)}\;(J\text{-comp})}{[a:\mathbb{C}^{\text{op}},b:\mathbb{C}]\;f:\hom_{\mathbb{C}}(a,b),k:P(\overline{a})\vdash \alpha[\text{subst}_P[f,k]]=\text{subst}_Q[f,\alpha[k]]:Q(b)}\;(J\text{-eq})$$

Example 3.8 (Internal dinaturality for entailments). For any $[x:\mathbb{C}]\;P(\overline{x},x)\vdash\alpha:Q(\overline{x},x)$, an internal version of (di)naturality for entailments, as in Definition 4.2, holds via (J-comp):

$$\dfrac{\dfrac{}{\begin{array}{r}[z:\mathbb{C}]\;k:P(\overline{z},z)\vdash \text{subst}_Q[(\text{refl}_z,\text{refl}_z),[\alpha[\text{subst}_P[(\text{refl}_z,\text{refl}_z),k]]]]\\ =\text{subst}_Q[(\text{refl}_z,\text{refl}_z),[\alpha[\text{subst}_P[(\text{refl}_z,\text{refl}_z),k]]]]:Q(\overline{z},z)\end{array}}\;(J\text{-comp})}{\begin{array}{r}[a:\mathbb{C}^{\text{op}},b:\mathbb{C}]\;f:\hom_{\mathbb{C}}(a,b),k:P(\overline{b},\overline{a})\vdash \text{subst}_Q[(\text{refl}_b,f),[\alpha[\text{subst}_P[(f,\text{refl}_a),k]]]]\\ =\text{subst}_Q[(f,\text{refl}_a),[\alpha[\text{subst}_P[(\text{refl}_b,f),k]]]]:Q(a,b)\end{array}}\;(J\text{-eq})$$

We elucidate more in detail why the above sequence of cuts is valid in Appendix H.

We show in Examples B.2 to B.4 how natural transformations between *terms* can be captured using ends [52, 1.4.1]. We show the identity natural, composition of naturals, and internal naturality.

3.1 On the adjoint formulation

We elaborate how the adjoint formulation, i.e., the fact that rules are formulated as bijections of entailments, differs from the standard type-theoretical presentation of connectives in the style of natural deduction or sequent calculus [56, 5.1.6]. Since in both of these systems cut is either derivable or admissible, we cannot recover the usual rules for introduction/elimination for quantifiers and implication, since in the semantics this would enable us to compose any two entailments/dinatural transformations. We give an example of introduction/elimination-like rules derivable from the adjoint formulation for (co)ends in Example 3.9.

Example 3.9 (Rules for (co)ends with terms). The following derivations capture an elimination rule for ends and, dually, an introduction rule for coends using a concrete diterm $\Gamma^{\mathrm{op}}, \Gamma \vdash F : \mathbb{C}$:

$$\dfrac{\dfrac{[\Gamma]\ \Phi \vdash \alpha : \int_{x:\mathbb{C}} P(\overline{x}, x)}{[x : \mathbb{C}, \Gamma]\ \Phi \vdash \mathsf{end}^{-1}(\alpha) : P(\overline{x}, x)}\ (\mathsf{end}^{-1})}{[\Gamma]\ \Phi \vdash F^{*}(\mathsf{end}^{-1}(\alpha)) : P(F, F)}\ (\mathsf{idx}) \qquad \dfrac{\dfrac{[\Gamma]\ k : \int^{x:\mathbb{C}} P(\overline{x}, x), \Phi \vdash \alpha : Q}{[x : \mathbb{C}, \Gamma]\ k : P(\overline{x}, x), \Phi \vdash \mathsf{coend}^{-1}(\alpha) : Q}\ (\mathsf{coend}^{-1})}{[\Gamma]\ k : P(F, F), \Phi \vdash F^{*}(\mathsf{coend}^{-1}(\alpha)) : Q}\ (\mathsf{idx})$$

We can recover the the projection and injection maps of (co)ends (i.e., the "(co)units" of the adjoint formulation) by picking $Q := \int^{x:\mathbb{C}} P(\overline{x}, x)$, $\Phi := \int_{x:\mathbb{C}} P(\overline{x}, x), \Phi'$ and $\alpha :=$ (var) as follows:

$$\dfrac{}{[\Gamma]\ k : \int_{x:\mathbb{C}} P(\overline{x}, x), \Phi' \vdash F^{*}(\mathsf{end}^{-1}(k)) : P(F, F)} \qquad \dfrac{}{[\Gamma]\ k : P(F, F), \Phi \vdash F^{*}(\mathsf{coend}^{-1}(k)) : \int^{x:\mathbb{C}} P(\overline{x}, x)}$$

The crucial aspect is that we cannot derive the above introduction/elimination rules where, instead, the end appears on the left, or the coend on the right: these would be the remaining rules for the quantifiers of *sequent calculus*, and hence full cut would be admissible. In particular we only recover $\forall_R$ and $\exists_L$, but not $\forall_L$ and $\exists_R$, using the terminology of [56, 5.1.8]. We formally prove the non-admissibility of an unrestricted cut rule in Theorem 5.3.

In standard accounts of logic, the adjoint-form is equivalent to the usual introduction and elimination rules for connectives, but only in the presence of *cut* [43, 4.1.8]. Hence, in our setting we can recover the usual rules only in contexts that are sufficiently *natural* to allow for cuts to be applied. We give an example of this situation in Example 3.10 to derive introduction/elimination-like rules for existentials in the style of natural deduction [56, 5.1.6], and derive in Example 3.11 transitivity of implication (which directly translates to an elimination rule).

Example 3.10 (Natural deduction-style rules for coends). The following derivations capture rules where coends are on the right of the turnstile: an elimination rule, an introduction rule with a concrete term $\Delta \vdash F : \mathbb{C}$ (not a *di*term), and an introduction rule with two variables $x : \mathbb{C}^{\mathrm{op}}, y : \mathbb{C}$:

$$\dfrac{\begin{array}{c}[\Gamma, d{:}\Delta]\ \Phi(d) \vdash \int^{x:\mathbb{C}} P(\overline{x}, x, d) \\ [\Gamma, z : \mathbb{C}, d{:}\Delta]\ P(\overline{z}, z, d), \Phi(d) \vdash Q(d)\end{array}}{[\Gamma, d : \Delta]\ \Phi(d) \vdash Q(d)} \qquad \dfrac{[\Gamma, d{:}\Delta]\ \Phi(d) \vdash Q(F(d), d)}{[\Gamma, d{:}\Delta]\ \Phi(d) \vdash \int^{x:\mathbb{C}} Q(x, d)} \qquad \dfrac{[\Gamma, x{:}\mathbb{C}^{\mathrm{op}}, y{:}\mathbb{C}]\ \Phi(x, y) \vdash R(x, y)}{[\Gamma]\ \Phi(x, y) \vdash \int^{z:\mathbb{C}} R(\overline{z}, z)}$$

Note that the variables of Δ are always used naturally, and P, Q, R do not depend on Γ. F cannot be a diterm since $Q(F(\overline{x}, x))$ would make the top entailment dinatural in the variables of Δ. We report complete derivations for these rules in Appendix C.

Example 3.11 (Transitivity of implication). Implication is transitive *in natural contexts*, with $[\Gamma]\ \Phi$:

$$\dfrac{\dfrac{[a : \mathbb{C}]\ \Phi \vdash \alpha : P(\overline{a}) \Rightarrow Q(a)}{[a : \mathbb{C}]\ P(a), \Phi \vdash \exp^{-1}(\alpha) : Q(a)}\ (\exp^{-1}) \qquad \dfrac{[a : \mathbb{C}]\ \Phi \vdash \beta : Q(\overline{a}) \Rightarrow R(a)}{[a : \mathbb{C}]\ Q(a), \Phi \vdash \exp^{-1}(\beta) : R(a)}\ (\exp^{-1})}{[a : \mathbb{C}]\ P(a), \Phi \vdash \alpha \,;\beta := \exp^{-1}(\beta)[\exp^{-1}(\alpha)] : R(a)}\ (\mathsf{cut\text{-}nat})$$

Polarized implication is in general not transitive, since, as we will see in Section 5, entailments are interpreted as dinaturals which do not compose in general; we show how in Theorem D.4 one can use implication and ends to internalize the set of all entailments/dinaturals.

3.2 Aspects of directed type theory

We investigate in this section other proof-theoretical aspects of our directed type theory: in particular we show why symmetry is not immediately derivable and how all rules for directed equality can be equivalently characterized as a single isomorphism.

Remark (Syntactic failure of symmetry for directed equality). *The restrictions in (J) illustrate why one* cannot *derive that directed equality is symmetric, i.e., obtain a general map*

$$[a : \mathbb{C}^{\mathsf{op}}, b : \mathbb{C}]\ e : \hom_{\mathbb{C}}(a, b) \vdash \mathsf{sym} : \hom_{\mathbb{C}}(\overline{b}, \overline{a}).$$

The equality $e : \hom_{\mathbb{C}}(a, b)$ cannot be contracted because $\overline{a}$ appears in the conclusion contravariantly (similarly with $\overline{b}$), whereas (J) requires that the conclusion only has covariant *occurrences of the variables being contracted.*

The remark above merely illustrates why it is not derivable *from the syntactic restriction.* We show in Theorem 5.2 that the existence of a countermodel implies that it is not admissible in general.

As in the symmetric case, the rule for directed equality elimination is actually an isomorphism, and asking (J) to be an isomorphism fully characterizes all the rules of directed equality [43, 3.2.3] (in the presence of the structural rules (cut-nat) and (var)):

Theorem 3.12 (Directed J as isomorphism). *Rule (J) is an isomorphism, and the inverse map is given by $J^{-1}(h) := h[\mathsf{refl}_{\mathbb{C}}]$ using (cut-nat) and (refl). Moreover, (J-eq) is logically equivalent to the rule $J(J^{-1}(\alpha)) = \alpha$ in the equational theory for every α.*

Proof. The computation rule states precisely that $J^{-1}(J(\alpha)) = \alpha$. To show $J(J^{-1}(\alpha)) = \alpha$, we instantiate (J-eq) with $\alpha := J(\beta[\mathsf{refl}_{\mathbb{C}}])$ and use (J-comp) in the hypothesis, i.e., $J(\beta[\mathsf{refl}_{\mathbb{C}}])[\mathsf{refl}_{\mathbb{C}}] = \beta[\mathsf{refl}_{\mathbb{C}}]$, to obtain $J(\beta[\mathsf{refl}_{\mathbb{C}}]) = \beta$ as desired. We show that $J(J^{-1}(\alpha)) = \alpha$ implies (J-eq): the hypothesis in (J-eq) states exactly $J^{-1}(\alpha) = J^{-1}(\beta)$, hence $\alpha = \beta$ by applying J on both sides. □

Theorem 3.13 ($J^{-1} \iff \mathsf{refl}$). *Rule (refl) is logically equivalent to (J^{-1}).*

Proof. Clearly (refl) implies (J^{-1}) by definition. Rule (refl) follows from (J^{-1}) in Theorem 3.12 by picking $P(a, b) := \hom(a, b)$ and using the projection (var) to return the hypothesis $e : \hom_{\mathbb{C}}(a, b)$ as the bottom side map h, obtaining $\mathsf{refl}_{\mathbb{C}} := J^{-1}(e)$. We leave the proof that the computation rule $J(h)[\mathsf{refl}_{\mathbb{C}}] = h$ holds in Appendix E. □

The following derivations illustrate how dinaturality, intuitively, allows us to "ignore" polarity in the contexts of predicates, i.e., one can equivalently consider a *contravariant* variable of type $\mathbb{C}$ as a *covariant* variable of type $\mathbb{C}^{\mathsf{op}}$, and viceversa.

Theorem 3.14 (op of entailments). *The following rule is derivable:*

$$\frac{[x : \mathbb{C}, \Gamma]\ \Phi(\overline{x}, x) \vdash \alpha : P(\overline{x}, x)}{[x : \mathbb{C}^{\mathsf{op}}, \Gamma]\ \Phi^{x \mapsto \mathsf{op}}(x, \overline{x}) \vdash \alpha^{x \mapsto \mathsf{op}} : P^{x \mapsto \mathsf{op}}(x, \overline{x})}$$

Proof. Follows by reindexing (idx) with the *"negative projection" diterm* $\overline{x} : \mathbb{C}, x : \mathbb{C}^{\mathsf{op}} \vdash \overline{x} : \mathbb{C}$. The predicate obtained by substituting this term in P coincides (metatheoretically) with $P^{x \mapsto \mathsf{op}}$. This reindexing is involutive in the sense that $(\alpha^{x \mapsto \mathsf{op}})^{x \mapsto \mathsf{op}} = \alpha$ in the equational theory. □

In particular, the above derivation allows us to *derive* different versions of (J) which adopt one or the other convention: for example (J) could be stated by requiring $a : \mathbb{C}$ (rather than $\mathbb{C}^{\mathsf{op}}$) but then ask for contravariance of a in the conclusion and covariance in Φ. The formulation chosen in (J) with $a : \mathbb{C}^{\mathsf{op}}, b : \mathbb{C}$ is simpler to state in terms of "correct" and "incorrect" appearances and emphasizes how the two variables play different asymmetric roles.

The following derivation shows how dinaturality allows us to capture a sort of "mixed-variance reindexing" $\mathbb{C} \to \mathbb{C}^{op} \times \mathbb{C}$, since even variables with different polarities can be identified together.

Theorem 3.15 (Dinatural collapse). *The following rule is derivable:*

$$\frac{[x : \mathbb{C}^{op}, y : \mathbb{C}, \Gamma]\ \Phi(\overline{x}, x, \overline{y}, y) \vdash \alpha : P(\overline{x}, x, \overline{y}, y)}{[z : \mathbb{C}, \Gamma]\ \Phi(z, \overline{z}, \overline{z}, z) \vdash \alpha^{x,y \mapsto z} : P(z, \overline{z}, \overline{z}, z)}$$

Proof. Follows by reindexing (idx) with the *"identity" diterm* $\overline{x} : \mathbb{C}^{op}, x : \mathbb{C} \vdash \langle \overline{x}, x \rangle : \mathbb{C}^{op} \times \mathbb{C}$. □

The dinatural collapse operation can be used to "downgrade" natural transformations to dinatural transformations, which no longer compose; since we check for naturality syntactically, this allows for a situation in which two (dinatural) entailments do not compose in the syntax despite composing in the semantics (since the map being constructed remains unaltered).

Remark (Collapse loses compositionality). *We illustrate how dinatural collapse can make an entailment no longer composable. Recall the composition map* $\mathsf{comp}[f, g] := J(g)$ *from Example 3.1: then, the following entailments are not composable in the syntax, since both* $\mathsf{comp}^{a,b \mapsto z}$ *and* refl *are dinatural in z; however,* $\mathsf{comp}[\mathsf{refl}_z, k]$ *is a valid application of (cut-nat):*

$$\frac{}{[z : \mathbb{C}]\ \Phi \vdash \mathsf{refl} : \hom_{\mathbb{C}}(\overline{z}, z)} \qquad \frac{\overline{[a : \mathbb{C}^{op}, b, c : \mathbb{C}]\ \hom_{\mathbb{C}}(a, b), \hom_{\mathbb{C}}(\overline{b}, c) \vdash \mathsf{comp} : \hom_{\mathbb{C}}(a, c)}}{[z, c : \mathbb{C}]\ \hom_{\mathbb{C}}(\overline{z}, z), \hom_{\mathbb{C}}(\overline{z}, c) \vdash \mathsf{comp}^{a,b \mapsto z} : \hom_{\mathbb{C}}(\overline{z}, c)}$$

Note that one can *apply* comp *to a constant dinatural* $[\,]\ \bullet \vdash \alpha : \hom_{\mathbb{C}}(A, A)$ *that selects some endomorphism for a concrete constant* $[\,] \vdash A : \mathbb{C}$*, since* α *would be natural in the empty context.*

We elucidate using (exp) why the exponential object in the category of presheaves and *natural transformations* is non-trivial [50, 6.3.20], and is not the pointwise hom in Set.

Remark (Exponentials for naturals). *Given an entailment which is fully covariant in x (i.e., a natural transformation) for predicates* $[x : \mathbb{C}]\ F(x), G(x), H(x)$*, by directly applying (exp),*

$$\frac{[x : \mathbb{C}]\ F(x) \times G(x) \vdash H(x)}{[x : \mathbb{C}]\qquad G(x) \vdash F(\overline{x}) \Rightarrow H(x)}\ (\text{exp})$$

one has a natural transformation on top, but the bottom family of arrows is dinatural *in x.*

We show in Example 6.2 how (exp) and the rules for directed equality can be used to give a logical proof that the usual definition of exponential for presheaves is indeed the correct one.

4 Dinaturality

We recall some preliminary facts about dinatural transformations and (co)ends in order to present the semantics of our type theory. We will often abbreviate the term dinatural transformations simply as "dinaturals", and ordinary natural transformations as "naturals".

Definition 4.1 (Dipresheaves and difunctors). Consider the (strict) comonad $-^{\diamond} : \mathsf{Cat} \to \mathsf{Cat}$ defined by $\mathbb{C} \mapsto \mathbb{C}^{op} \times \mathbb{C}$, where the counit is given by projecting and comultiplication by duplicating and swapping. A *dipresheaf* is simply a functor $\mathbb{C}^{\diamond} \to \mathsf{Set}$, i.e. a functor $\mathbb{C}^{op} \times \mathbb{C} \to \mathsf{Set}$.

We always denote composition diagrammatically, i.e., $f ; g : a \to c$ for $f : a \to b, g : b \to c$.

Definition 4.2 (Dinatural transformation [26]). Given functors $F, G : \mathbb{C}^{op} \times \mathbb{C} \to \mathbb{D}$, a *dinatural transformation* $\alpha : F \xrightarrow{\bullet\bullet} G$ is a family of arrows $\alpha_x : F(x, x) \longrightarrow G(x, x)$ indexed by objects $x : \mathbb{C}$ such that $\forall a, b : \mathbb{C}$, and $f : a \to b$ the following equation between arrows $F(b, a) \to G(a, b)$ holds:

$$F(\mathsf{id}_b, f) ; \alpha_b ; G(f, \mathsf{id}_b) = F(f, \mathsf{id}_a) ; \alpha_a ; G(\mathsf{id}_b, f).$$

Lemma 4.3 (Dinaturals generalize naturals [26]). *A natural transformation* $\alpha : F \to G$ *for* $F, G : \mathbb{C} \to \mathbb{D}$ *equivalently corresponds with a dinatural* $\alpha : (\pi_2 \,;\, F) \xrightarrow{\bullet\bullet} (\pi_2 \,;\, G) : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{D}$.

The pointwise composition of two dinatural transformations is not necessarily dinatural (see [30, 55]), but dinaturals always compose with naturals on both the left and right side:

Lemma 4.4 (Dinaturals compose with naturals [26]). *Given a dinatural transformation* $\gamma : F \xrightarrow{\bullet\bullet} G$ *and natural transformations* $\alpha : F' \to F, \beta : G \to G'$ *for* $F, F', G, G' : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathrm{Set}$, *the map* $\alpha \,;\, \gamma \,;\, \beta : F' \xrightarrow{\bullet\bullet} G'$ *defined by* $(\alpha \,;\, \gamma \,;\, \beta)_x := \alpha_{xx} \,;\, \gamma_x \,;\, \beta_{xx}$ *is dinatural.*

Non-compositionality of dinaturals is an intrinsic property of *directed proof-relevant* type theory, since in the groupoidal case they all compose (in the proof-irrelevant case, where Set is replaced by the preorder $\mathbf{I} := \{0 \to 1\}$, dinaturals compose trivially since there is no hexagon to check):

Theorem 4.5 (Dinaturals in groupoids compose). *Given a groupoid* $\mathbb{C}$ *and a category* $\mathbb{D}$ *for functors* $F, G, H : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{D}$, *any two dinaturals* $\alpha : F \xrightarrow{\bullet\bullet} G, \beta : G \xrightarrow{\bullet\bullet} H$ *compose.*

The fundamental idea behind all rules for directed equality is given by the following elementary result, which connects dinatural transformations in Set with a corresponding natural one:

Theorem 4.6 (Dinaturals and hom-naturals). *For any* $P, Q : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathrm{Set}$, *there is a bijection between set of dinatural transformations* $P \xrightarrow{\bullet\bullet} Q$ *and certain natural transformations between functors* $\mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathrm{Set}$, *as follows:*

$$\frac{\alpha_x : P(\overline{x}, x) \xrightarrow{\bullet\bullet} Q(\overline{x}, x)}{\gamma_{ab} : \hom(a, b) \longrightarrow P^{\mathrm{op}}(b, a) \Rightarrow Q(a, b)}$$

Proof. We describe the maps in both directions:

- (⇓) Given a dinatural $\alpha : P \xrightarrow{\bullet\bullet} Q$ and a morphism $f : \hom(a, b)$, the map $P(b, a) \to Q(a, b)$ corresponds precisely with the sides of the equation given in Definition 4.2 for dinaturality, which is obtained by applying the functorial action of P and Q.
- (⇑) Take $a = b$ and precompose with $\mathrm{id}_a \in \hom(a, a)$.

The fact that this is an isomorphism follows from the (di)naturality of both sets of maps. Note the similarity between the above argument and the proof of the Yoneda lemma, where the two central ideas are precisely applying the functorial action and instantiating at id, with the isomorphism following from (di)naturality. □

We now recall definitions for the semantics of (co)ends, later used to give semantics to quantifiers.

Definition 4.7 ((Co)wedges for P [52, 1.1.4]). Given $P : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{D}$, a *wedge for* P is a pair object/dinatural $(X : \mathbb{D}, \alpha : \mathrm{K}_X \xrightarrow{\bullet\bullet} P)$, where K_X is the constant functor in X. A *wedge morphism* $(X, \alpha) \to (Y, \alpha')$ is an $f : X \to Y$ of $\mathbb{D}$ such that $\forall c : \mathbb{C}, \alpha_c = f \,;\, \alpha'_c$. A *cowedge* is a wedge in $\mathbb{D}^{\mathrm{op}}$, denoting the categories of (co)wedges as $\mathrm{Wedge}(P), \mathrm{Cowedge}(P)$.

Definition 4.8 ((Co)ends [52, 1.1.6]). Given a functor $P : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{D}$, the *end* of P is defined to be the terminal object of $\mathrm{Wedge}(P)$, whose object in $\mathbb{D}$ is denoted as $\int_{x:\mathbb{C}} P(\overline{x}, x)$. Dually, the *coend* of P is the initial object of $\mathrm{Cowedge}(P)$, denoted similarly as $\int^{x:\mathbb{D}} P(\overline{x}, x)$. The integral symbol acts as a binder, in the sense that "$\int_{c:\mathbb{C}} P(c, c)$" and "$\int_{x:\mathbb{C}} P(x, x)$" are ($\alpha$-)equivalent; moreover, P can depend on many parameters, e.g., if $P : (\mathbb{A}^{\mathrm{op}} \times \mathbb{A}) \times (\mathbb{B}^{\mathrm{op}} \times \mathbb{B}) \to \mathbb{D}$ then $\int_{b:\mathbb{B}} P(\overline{a}, a, \overline{b}, b) : \mathbb{A}^{\mathrm{op}} \times \mathbb{A} \to \mathbb{D}$. (Co)ends exist when $\mathbb{D}$ is (co)complete [52].

5 Semantics

We now describe the categorical semantics of our directed type theory: the main idea behind categorical semantics is that we define functions that associate a certain mathematical object to each derivation tree, inductively. Whenever present, the symbol (🐓) links to the Agda formalization of the semantic interpretation of each rule.

The semantics for types, contexts, variables, terms, predicates and propositional contexts is given in Figure 12. The equality judgments associated to these are interpreted in a straightforward way, which we omit from this presentation; such equalities are only used to take care of involutions and the equational theory of terms, for which we therefore give a *strict* semantics: equality of types and contexts is interpreted as *isomorphisms* of categories, term equality is strict isomorphism of functors. Equality of predicates is similarly trivial since it only inherits congruence rules from the previous equality judgments.

The main rules of our type theory are those of entailments, for which we describe in detail the intuition behind the semantics of each rule and its soundness in dinatural transformations.

$$
\begin{aligned}
&[\![-]\!] : \{-\ \mathsf{type}\} \to \mathsf{Cat}\\
&[\![\mathbb{C}^{\mathsf{op}}]\!] := [\![\mathbb{C}]\!]^{\mathsf{op}}\\
&[\![\mathbb{C}\times\mathbb{D}]\!] := [\![\mathbb{C}]\!]\times[\![\mathbb{D}]\!]\\
&[\![[\mathbb{C},\mathbb{D}]]\!] := [[\![\mathbb{C}]\!],[\![\mathbb{D}]\!]]\\
&[\![\top]\!] := \top\\[1ex]
&[\![-]\!] : \{-\ \mathsf{ctx}\} \to \mathsf{Cat}\\
&[\![[]]\!] := \top\\
&[\![\Gamma^{\mathsf{op}}]\!] := [\![\Gamma]\!]^{\mathsf{op}}\\
&[\![\Gamma,\mathbb{C}]\!] := [\![\Gamma]\!]\times[\![\mathbb{C}]\!]
\end{aligned}
$$

$$
\begin{aligned}
&[\![-]\!]^v : \{\Gamma \ni - : \mathbb{C}\} \to [[\![\Gamma]\!],[\![\mathbb{C}]\!]]\\
&[\![\Gamma, x:\mathbb{C} \ni x:\mathbb{C}]\!]^v := \pi_2\\
&[\![\Gamma, y:\mathbb{D} \ni y:\mathbb{C}]\!]^v := \pi_1 \,;\, [\![y]\!]^v\\[1ex]
&[\![-]\!] : \{\Gamma \vdash - : \mathbb{C}\} \to [[\![\Gamma]\!],[\![\mathbb{C}]\!]]\\
&[\![x]\!] := [\![x]\!]^v\\
&[\![t^{\mathsf{op}}]\!] := [\![t]\!]^{\mathsf{op}}\\
&[\![\langle s,t\rangle]\!] := \langle [\![s]\!],[\![t]\!]\rangle\\
&[\![\pi_1(p)]\!] := [\![p]\!] \,;\, \pi_1\\
&[\![\pi_2(p)]\!] := [\![p]\!] \,;\, \pi_2\\
&[\![s\cdot t]\!] := \langle [\![s]\!],[\![t]\!]\rangle \,;\, \mathsf{eval}\\
&[\![\lambda x.t(x)]\!] := \Lambda(t)
\end{aligned}
$$

$$
\begin{aligned}
&[\![-]\!] : \{[\Gamma] -\ \mathsf{prop}\} \to [[\![\Gamma]\!]^{\mathsf{op}}\times[\![\Gamma]\!],\mathsf{Set}]\\
&[\![\top]\!] := \lambda\overline{\gamma},\gamma.\top_{\mathsf{Set}}\\
&[\![P\times Q]\!] := \langle [\![P]\!],[\![Q]\!]\rangle \,;\, \times_{\mathsf{Set}}\\
&[\![P\Rightarrow Q]\!] := \langle [\![P]\!],[\![Q]\!]\rangle \,;\, \Rightarrow_{\mathsf{Set}}\\
&[\![\hom_{\mathbb{C}}(s,t)]\!] := \langle [\![s]\!],[\![t]\!]\rangle \,;\, \hom_{\mathbb{C}}\\
&[\![\textstyle\int_{x:\mathbb{C}} P(\overline{x},x)]\!] := \lambda\overline{\gamma},\gamma.\textstyle\int_{x:\mathbb{C}} P(\overline{x},x,\overline{\gamma},\gamma)\\
&[\![\textstyle\int^{x:\mathbb{C}} P(\overline{x},x)]\!] := \lambda\overline{\gamma},\gamma.\textstyle\int^{x:\mathbb{C}} P(\overline{x},x,\overline{\gamma},\gamma)\\[1ex]
&[\![-]\!] : \{-\ \mathsf{propctx}\} \to [[\![\Gamma]\!]^{\mathsf{op}}\times[\![\Gamma]\!],\mathsf{Set}]\\
&[\![\bullet]\!] := \lambda\overline{\gamma},\gamma.\top_{\mathsf{Set}}\\
&[\![\Phi,P]\!] := \langle [\![\Phi]\!],[\![P]\!]\rangle \,;\, \times_{\mathsf{Set}}
\end{aligned}
$$

Fig. 12. Semantics for the main judgments of directed dinatural type theory.

Theorem 5.1 (Soundness in dinatural transformations). (🐓) *Each rule in Figure 11 is validated using the semantics in categories, functors, dipresheaves, dinatural transformations. Inference rules are interpreted by functions between sets of dinaturals; these are isomorphisms when double-lines appear. Moreover, every function is* natural *in all the dipresheaves (both predicates and propositional contexts) that appear in the rule.*

We unpack this theorem by validating and describing the intuition behind each rule, using semantic brackets $[\![-]\!]$ to indicate the semantic object denoted by each constructor.

- **Structural rules.** (🐓) Rule (var) is interpreted as the dinatural which projects away the predicate P. Moreover, (wk) and (contr) state that dinaturals always compose on the left with, respectively, the projections and the diagonal map in Set.
- **Products.** (🐓) Dinaturals validate the interpretation of conjunction in (prod) via the pointwise product of dipresheaves in Set; the bottom sequent indicates the product of sets of dinaturals.
- **Polarized implication.** (🐓) Contrary to naturals and presheaves [50], dinaturals can be curried directly via the (exp) rule by currying each component of α in Set. In the semantics, the metatheoretical operation Example 2.10 corresponds to swapping arguments in a dipresheaf.
- **Reindexing with functors as terms.** (🐓) Dinaturals can always be "reindexed" by plugging functors in each index of the component, preserving dinaturality.

- **Cuts naturals-dinaturals.** (👋) The two restricted cut rules (cut-din), (cut-nat) correspond precisely to Lemma 4.4. Intuitively, both rules are stated in such a way that the dipresheaf P (in the middle of the composition) only contains *natural* occurrences of variables. The use of Γ in Φ, Q is unproblematic since one can suitably take the (co)end over Γ to "hide" these variables and compose naturals together. Associativity, unitality and coherence in Figure 15 are immediate. The dinatural-into-natural rule (cut-nat) essentially corresponds to vertical composition in Prof as a virtual equipment [24, 60]: in this type theory, however, contravariant occurrences $\overline{a}, \overline{b}$ are allowed to appear in the *same* predicate $P(\overline{a}, \overline{b})$, but in the double-categorical setting they must be split as $P(..., a), Q(\overline{a}, b), R(\overline{b}, ...)$. Note that composing a natural with a dinatural yields a *di*natural, hence the resulting map is always collapsed via Theorem 3.2, e.g., in (cut-nat-id$_l$), (cut-din-id$_r$).
- **Directed equality introduction.** (👋) The rule (refl) states reflexivity of directed equality, and is validated semantically by $\alpha_x(h) := \mathrm{id}_x$. Dinaturality holds since $\forall f : a \to b, f \mathbin{;} \mathrm{id}_b = \mathrm{id}_a \mathbin{;} f$.
- **Directed equality elimination.** (👋) This rule and its syntactic restriction comes precisely from Theorem 4.6: in the bottom side of the isomorphism, the dipresheaf P is curried on the left of the turnside *but inverting the polarity of* a, b. This is precisely the propositional context of (J). Hence, the restriction behind (J) comes from the naturality of the bottom map. Explicitly, given a dinatural h, the dinatural $J(h)$ is defined as follows for indices $a : [\![\mathbb{C}]\!], b : [\![\mathbb{C}^{\mathrm{op}}]\!], x : [\![\Gamma]\!]$:

$$J(h)_{abx} := \lambda e, k.([\![\Phi]\!](\mathrm{id}_b, e, \mathrm{id}_x, \mathrm{id}_x) \mathbin{;} h_{bx} \mathbin{;} [\![P]\!](e, \mathrm{id}_b, \mathrm{id}_x, \mathrm{id}_x))(k).$$

  The computation rule clearly holds when $a = b = z$ and $e = \mathrm{id}_z$, without the need for dinaturality.
- **Dependent** hom **elimination.** (👋) As shown in Theorem 3.12, the fact that J is an isomorphism characterizes directed equality. In particular, dependent equality elimination is the $J(J^{-1}(\alpha)) = \alpha$ direction, which uses naturality in the proof just like the Yoneda lemma [50, 4.2].
- **(Co)ends.** (👋) The rules for (co)ends (end) and (coend) express an adjoint-like (up to the non-composition of dinaturals) correspondence $\int^{\mathbb{A}[\mathbb{C}]} \dashv \pi^*_{\mathbb{A}[\mathbb{C}]} \dashv \int_{\mathbb{A}[\mathbb{C}]}$ between the weakening functor $\pi^*_{\mathbb{A}[\mathbb{C}]} : [\mathbb{C}^\diamond, \mathrm{Set}] \to [\mathbb{A}^\diamond \times \mathbb{C}^\diamond, \mathrm{Set}]$ and the functors $\int^{\mathbb{A}[\mathbb{C}]}, \int_{\mathbb{A}[\mathbb{C}]} : [\mathbb{A}^\diamond \times \mathbb{C}^\diamond, \mathrm{Set}] \to [\mathbb{C}^\diamond, \mathrm{Set}]$ sending dipresheaves to their (co)end in A. Semantically, these are simply the bijective correspondences between (co)wedges and morphisms (out of) into (co)ends, but parameterized by an additional context of variables Γ. Quantifiers in categorical logic typically have to satisfy additional requirements in order to faithfully model logical operations: the Beck-Chevalley condition [43, 1.9.4] states that "quantifiers commute with substitution", and the Frobenius condition [43, 1.9.12] logically corresponds to having an additional context Φ in rules for colimit-like connectives [43, 3.4.4], as in (coend). We show these technical conditions in Theorem F.1.

Theorem 5.2 (Symmetry is not admissible). *The statement of symmetry of directed equality in Remark 3.2 is not admissible in the type theory.*

Proof. Add to the signature the category $\mathbf{I} := \{0 \to 1\}$ with a unique non-invertible morphism. By soundness, the lack of symmetry in $\mathbf{I}$ implies that symmetry cannot be derived in general. □

The set of all dinaturals can be characterized as an end $\mathrm{Dinat}(P, Q) \cong \int_{x:\mathbb{C}} P(x, \overline{x}) \Rightarrow Q(\overline{x}, x)$; we prove this in Theorem D.4. We internalize this idea to show that full cut cannot be derived:

Theorem 5.3 (No full cut). *A cut rule where Φ, P, Q are fully unrestricted is* not *admissible.*

Proof. Assuming full cut, the adjoint formulation is equivalent to the rules of natural deduction of first-order logic, which allows one to derive the following map in the empty context:

$$\frac{}{[\,]\ \int_{x:\mathbb{C}} P(x, \overline{x}) \Rightarrow Q(\overline{x}, x), \int_{x:\mathbb{C}} Q(x, \overline{x}) \Rightarrow R(\overline{x}, x) \vdash \int_{x:\mathbb{C}} P(x, \overline{x}) \Rightarrow R(\overline{x}, x)}$$

by soundness of the semantics, this corresponds to composing *all* dinatural transformations. □

6 Coend calculus via dinaturality

We show how the rules for directed equality and (co)ends can be used to give concise proofs with a distinctly logical flavor to several central theorems of category theory. The technique we use mirrors the way (co)end calculus is applied in practical settings (e.g., [15, 40, 74]) via a "Yoneda-like" series of *natural* isomorphisms of sets: to prove that two objects $A, B : \mathbb{C}$ are isomorphic, one can assume to have a generic object Φ and then apply a series of isomorphisms of sets *natural* in Φ to establish that $\mathbb{C}(\Phi, A) \cong \mathbb{C}(\Phi, B)$, from which $A \cong B$ follows by the fully faithfulness of the Yoneda embedding [15, 50]. The same technique can be used to show that *functors* are naturally isomorphic, as well as adjunctions, e.g., Examples 6.2 and 6.3. We now show our main examples, with additional derivations of (co)end calulus in Appendix D, which use Yoneda with Φ on the right side instead.

Remark (Yoneda technique and naturality). *All rules given in previous sections are* natural *in each of the dipresheaves involved. In the following series of examples no proof ever involves a "dinatural isomorphism", since it would not be possible to state the final isomorphism with cuts; natural isomorphisms between sets of* dinaturals *are only used as intermediate steps. We show in Appendix G a spelled-out example of this Yoneda technique in the equational theory by explicitly constructing the isomorphisms and using naturality of the adjoint-form rules (i.e., they commute with cuts).*

Example 6.1 ((co)Yoneda lemma). For any predicate/copresheaf $[x : \mathbb{C}]\ P(x)$ prop, and a predicate/copresheaf $[x : \mathbb{C}]\ \Phi(x)$ propctx acting as generic context, the following derivations capture the Yoneda lemma [52, Thm. 1] (using the characterization of naturals as an end) and coYoneda lemma [53, III.7, Theorem 1] (i.e., presheaves are isomorphic to a weighted colimit of representables)

$$\begin{array}{c} [a{:}\mathbb{C}]\ \Phi(a) \vdash \int_{x:\mathbb{C}} \hom_{\mathbb{C}}(a, \overline{x}) \Rightarrow P(x) \\ \hline\hline [a{:}\mathbb{C}, x{:}\mathbb{C}]\ \Phi(a) \vdash \hom_{\mathbb{C}}(a, \overline{x}) \Rightarrow P(x) \\ \hline\hline [a{:}\mathbb{C}, x{:}\mathbb{C}]\ \hom_{\mathbb{C}}(\overline{a}, x), \Phi(a) \vdash P(x) \\ \hline\hline [z : \mathbb{C}]\ \Phi(z) \vdash P(z) \end{array} \begin{array}{l} \text{(end)} \\ \text{(exp)} \\ (J) \end{array} \qquad \begin{array}{c} [a{:}\mathbb{C}]\ \int^{x:\mathbb{C}} \hom_{\mathbb{C}}(\overline{x}, a) \times P(x) \vdash \Phi(a) \\ \hline\hline [a{:}\mathbb{C}, x{:}\mathbb{C}]\ \hom_{\mathbb{C}}(\overline{x}, a) \times P(x) \vdash \Phi(a) \\ \hline\hline [z : \mathbb{C}]\ P(z) \vdash \Phi(z) \end{array} \begin{array}{l} \text{(coend)} \\ (J) \end{array}$$

Example 6.2 (Presheaves are cartesian closed). For any $[\mathbb{C}]\ A, B, \Phi$, the following derivation shows that the internal hom in the category of presheaves and naturals [50, 6.3.20] defined by $(A \Rightarrow B)(x) := \mathsf{Nat}(\hom(x, -) \times A, B)$ is indeed the correct one. We show here the tensor/hom adjunction:

$$\begin{array}{c} [x : \mathbb{C}]\ \Phi(x) \vdash (A \Rightarrow B)(x) := \mathsf{Nat}(\hom_{\mathbb{C}}(x, -) \times A, B) \\ = \int_{y:\mathbb{C}} \hom_{\mathbb{C}}(x, \overline{y}) \times A(\overline{y}) \Rightarrow B(y) \\ \hline\hline [x : \mathbb{C}, y : \mathbb{C}]\ \Phi(x) \vdash \hom_{\mathbb{C}}(x, \overline{y}) \times A(\overline{y}) \Rightarrow B(y) \\ \hline\hline [x : \mathbb{C}, y : \mathbb{C}]\ A(y) \times \hom_{\mathbb{C}}(\overline{x}, y) \times \Phi(x) \vdash B(y) \\ \hline\hline [y : \mathbb{C}]\ A(y) \times \left(\int^{x:\mathbb{C}} \hom_{\mathbb{C}}(\overline{x}, y) \times \Phi(x)\right) \vdash B(y) \\ \hline\hline [y : \mathbb{C}]\ A(y) \times \Phi(y) \vdash B(y) \end{array} \begin{array}{l} \\ \\ \text{(end)} \\ \text{(exp)} \\ \text{(coend)} \\ \text{(coYoneda)} \end{array}$$

We precompose with the (coYoneda) isomorphism given in Example 6.1 (which is a *natural* isomorphism). Note that (J) cannot be applied immediately since y appears positively in context in $A(y)$, whereas it should be negative to identify it with x. The above derivation is a simple application of our rules via dinaturality, but it is unclear how it can be captured using the proarrow equipment approach of [60, 85] as an abstract property of Prof, due to the repetition of variables $y, \overline{y}$.

Example 6.3 (Pointwise formula for right Kan extensions). Using our rules, we give a logical proof that the functor $\mathsf{Ran}_F : [\mathbb{C}, \mathsf{Set}] \to [\mathbb{D}, \mathsf{Set}]$ sending (co)presheaves to their Kan extensions along

$F : \mathbb{C} \to \mathbb{D}$ computed via ends [52, 2.3.6] is right adjoint to precomposition $(F\,;-) : [\mathbb{D}, \mathrm{Set}] \to [\mathbb{C}, \mathrm{Set}]$. We again precompose with the (coYoneda) isomorphism, which we reindex implicitly with F. Note the similarity between this derivation and the argument given in [71, 5.6.6] to compute adjoints in a general doctrine. For any $[x : \mathbb{C}]\ P(x)$, $[y : \mathbb{D}]\ \Phi(y)$, a functor/term $F : \mathbb{C} \to \mathbb{D}$:

$$
\begin{array}{cl}
[y : \mathbb{D}]\ \Phi(y) \vdash (\mathrm{Ran}_F P)(y) := \int_{x:\mathbb{C}} \hom_{\mathbb{D}}(y, F(\overline{x})) \Rightarrow P(x) & \\
\hline\hline
[x : \mathbb{C}, y : \mathbb{D}]\ \Phi(y) \vdash \hom_{\mathbb{D}}(y, F(\overline{x})) \Rightarrow P(x) & (\text{end}) \\
\hline\hline
[x : \mathbb{C}, y : \mathbb{D}]\ \hom_{\mathbb{D}}(\overline{y}, F(x)) \times \Phi(y) \vdash P(x) & (\text{exp}) \\
\hline\hline
[x : \mathbb{C}]\ \int^{y:\mathbb{D}} \hom_{\mathbb{D}}(\overline{y}, F(x)) \times \Phi(y) \vdash P(x) & (\text{coend}) \\
\hline\hline
[y : \mathbb{C}]\ \Phi(F(x)) \vdash P(x) & (\text{coYoneda})
\end{array}
$$

Example 6.4 (Fubini rule for ends). For convenience we only show the case for ends. For $[\,]\ \Phi$ propctx in the empty context (i.e., just an object $[\![\Phi]\!] : \mathrm{Set}$) and $[\mathbb{C}, \mathbb{D}]\ P$ prop the following are all equivalent thanks to the fact that certain structural properties of contexts hold by cartesianness of Cat.

$$
\begin{array}{cl}
[\,]\ \Phi \vdash \int_{x:\mathbb{C}} \int_{y:\mathbb{D}} P(\overline{x}, x, \overline{y}, y) & \\
\hline\hline
[x : \mathbb{C}]\ \Phi \vdash \int_{y:\mathbb{D}} P(\overline{x}, x, \overline{y}, y) & (\text{end}) \\
\hline\hline
[x : \mathbb{C}, y : \mathbb{D}]\ \Phi \vdash P(\overline{x}, x, \overline{y}, y) & (\text{end}) \\
\hline\hline
[y : \mathbb{D}, x : \mathbb{C}]\ \Phi \vdash P(\overline{x}, x, \overline{y}, y) & (\text{structural property})
\end{array}
\qquad
\begin{array}{cl}
\cdots & \\
\hline\hline
[p : \mathbb{C} \times \mathbb{D}]\ \Phi \vdash P(\overline{p}, p) & (\text{structural property}) \\
\hline\hline
[y : \mathbb{D}]\ \Phi \vdash \int_{x:\mathbb{C}} P(\overline{x}, x, \overline{y}, y) & (\text{end}) \\
\hline\hline
[\,]\ \Phi \vdash \int_{y:\mathbb{D}} \int_{x:\mathbb{C}} P(\overline{x}, x, \overline{y}, y) & (\text{end}) \\
\hline\hline
[\,]\ \Phi \vdash \int_{p:\mathbb{C}\times\mathbb{D}} P(\overline{x}, x, \overline{y}, y) & (\text{end})
\end{array}
$$

Example 6.5 ($\Rightarrow$ resp. limits). Ends are limits [52], and functors $- \Rightarrow - : \mathrm{Set}^{\mathrm{op}} \times \mathrm{Set} \to \mathrm{Set}$ preserve them (ends/limits in $\mathrm{Set}^{\mathrm{op}}$, i.e., coends/colimits in Set). For $[]\ \Phi$ propctx, $[]\ Q$ prop, $[\mathbb{C}]\ P$ prop:

$$
\begin{array}{cl}
[\,]\ \Phi \vdash Q \Rightarrow \int_{x:\mathbb{C}} P(\overline{x}, x) & \\
\hline\hline
[\,]\ Q, \Phi \vdash \int_{x:\mathbb{C}} P(\overline{x}, x) & (\text{exp}) \\
\hline\hline
[x : \mathbb{C}]\ Q, \Phi \vdash P(\overline{x}, x) & (\text{end}) \\
\hline\hline
[x : \mathbb{C}]\ \Phi \vdash Q \Rightarrow P(\overline{x}, x) & (\text{exp}) \\
\hline\hline
[\,]\ \Phi \vdash \int_{x:\mathbb{C}} (Q \Rightarrow P(\overline{x}, x)) & (\text{end})
\end{array}
\qquad
\begin{array}{cl}
[\,]\ \Phi \vdash (\int^{x:\mathbb{C}} P(\overline{x}, x)) \Rightarrow Q & \\
\hline\hline
[\,]\ (\int^{x:\mathbb{C}} P(\overline{x}, x)), \Phi \vdash Q & (\text{exp}) \\
\hline\hline
[x : \mathbb{C}]\ P(\overline{x}, x), \Phi \vdash Q & (\text{coend}) \\
\hline\hline
[x : \mathbb{C}]\ \Phi \vdash P(x, \overline{x}) \Rightarrow Q & (\text{exp}) \\
\hline\hline
[\,]\ \Phi \vdash \int_{x:\mathbb{C}} P(x, \overline{x}) \Rightarrow Q & (\text{end})
\end{array}
$$

7 Conclusions and future work

In this paper we showed how dinaturality is the key notion to give a simple and natural description to a first-order directed type theory where types are interpreted as (1-)categories and directed equality as hom-functors. Our type theory is powerful enough to express theorems about directed equality in a straightforward way, and to give a distinctly logical interpretation to well-known theorems in category theory by reinterpreting them under the light of directed type theory.

Dinaturality. The compositionality problem of dinatural transformations is a long-standing and famously difficult problem [75], which both the category theory and computer science communities have relatively left unexplored since their introduction in the 1970s [26, 27]. Our work gives a concrete motivation to further investigate this more than 50-years old mystery by connecting it to directed type theory. We conjecture that this connection could possibly hint to a deeper *directed homotopical* reason [28, 33] for why dinaturals fail to compose. Strong dinaturals [58, 65] are one possible approach to deal with non-compositionality, but they lack in expressivity, e.g., they are not closed in general [79]. Following Theorem 4.5, this non-compositionality is intrinsic to the directed proof-relevant setting, i.e., non-groupoidal categories. We leave investigating the relation between dinaturality and geometric models of $(\infty, 1)$-categories in the spirit of [34, 73, 84] for future work.

Type dependency. Our treatment of directed equality via dinaturality is a first step towards understanding the precise interplay of polarity, directedness and variance in fully dependent Martin-Löf type theory, especially with respect to how polarity of variables is influenced by their appearance in types, which we conjecture to be particularly non-trivial.

Initiality. The syntactic system presented in this paper could be axiomatized into a suitable initial object in a category of models that captures the behavior of variables in dinaturals and naturals (e.g., as in [75]): one possible approach could be to abstractly consider two classes of maps (dinaturals, naturals) and requiring such maps to interact as in Lemma 4.4.

Doctrines. All of our results can be specialized in the category of posets Pos rather than Cat, where dinaturals compose trivially and our work provides a "logic of posets", captured via a bona fide doctrine, at the cost of trivializing (co)ends with (co)products. This posetal case could be axiomatized in the style of the doctrinal approach [43, 54], with a notion of *directed doctrine* capturing the roles played by variance, the $-^{\mathrm{op}}$ involution, and (di)naturality. This would allow our syntactic rules to be organized in a well-known structure, with a suitable initiality result.

Internalizing Yoneda. The Yoneda technique for isomorphisms follows from "manually" using naturality of isomorphisms in the equational theory. One could also get this naturality for free by making the theory second-order with a universe of propositions Set and adding a directed univalence statement $\hom_{\mathrm{Set}}(A, B) \cong A \Rightarrow B$ (as in [4, 34, 84]): this would allow for implication to be represented as a directed equality, contractible with (J), and "synthetically" reproduce the same argument as in Example 3.8 by quantifying over all predicates involved.

Higher (co)end calculus. There are other conceptual examples of coend calculus which have not yet been interpreted in terms of directed equality: for instance, one should be able to express that composition maps exist *for all* categories $\mathbb{C}$: Cat, where this quantification can be expressed via a suitable pseudo-end in Cat [52, 7.1]; similarly, the category of elements of a functor, reminiscent of a Σ-type, can be given as the pseudo-coend $\mathsf{El}(F) \cong \int^{c:\mathbb{C}} c/\mathbb{C} \times F(c)$, where $c/\mathbb{C}$ is the coslice category and $F(c)$ is seen as a discrete category [52, 4.2.2]. These examples could be captured by considering the category of small categories Cat as a suitable universe of types [41].

Enrichment. We do not rely on specific properties of Set (viewed as the base of enrichment of Cat), other than cartesian closedness to have propositional implication/conjunction and the existence of (co)limits to express (co)ends. We conjecture that our analysis of dinaturals can be developed in more generality by taking enriched categories (over a sufficiently structured base of enrichment) as types, rather than simply categories (enriched over Set).

Implementation. We remark how an implementation of the metatheory of our type-theoretical system in a proof assistant is non-trivial, since one has to push $-^{\mathrm{op}}$ down into connectives and ensure that $(X^{\mathrm{op}})^{\mathrm{op}} \equiv X$ everywhere in the syntax: in types, contexts, terms, predicates, propositional contexts. This could be tackled in practice by using QITs [3] and the `--rewriting` feature of Agda [21] to simplify op whenever necessary. Another solution would be to have $-^{\mathrm{op}}$ only at the level of base types, and then derive $-^{\mathrm{op}}$ as a metatheoretical operation on full types; this has the disadvantage that $-^{\mathrm{op}}$ is not a primitve type former that one can explicitly manipulate in the syntax.

Acknowledgments

The authors thank the reviewers for their detailed suggestions and Paweł Sobociński for invaluable feedback on the presentation of this work. Loregian was supported by the Estonian Research Council grant PRG1210. Veltri was supported by the Estonian Research Council grant PSG749.

References

[1] Benedikt Ahrens, Paige Randall North, and Niels van der Weide. 2022. Semantics for two-dimensional type theory. In *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '22)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3531130.3533334

[2] Benedikt Ahrens, Paige Randall North, and Niels van der Weide. 2023. Bicategorical type theory: semantics and syntax. *Mathematical Structures in Computer Science* (Oct. 2023), 1–45. doi:10.1017/S0960129523000312

[3] Thorsten Altenkirch and Ambrus Kaposi. 2016. Type theory in type theory using quotient inductive types. In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '16)*. Association for Computing Machinery, New York, NY, USA, 18–29. doi:10.1145/2837614.2837638

[4] Thorsten Altenkirch and Jacob Neumann. 2024. Synthetic 1-Categories in Directed Type Theory. arXiv:2410.19520 [math] doi:10.48550/arXiv.2410.19520 arXiv:2410.19520.

[5] Kazuyuki Asada. 2010. Arrows are strong monads. In *Proceedings of the third ACM SIGPLAN workshop on Mathematically structured functional programming (MSFP '10)*. Association for Computing Machinery, New York, NY, USA, 33–42. doi:10.1145/1863597.1863607

[6] Kazuyuki Asada and Ichiro Hasuo. 2010. Categorifying Computations into Components via Arrows as Profunctors. *Electronic Notes in Theoretical Computer Science* 264, 2 (Aug. 2010), 25–45. doi:10.1016/j.entcs.2010.07.012

[7] Steve Awodey and Michael A. Warren. 2009. Homotopy theoretic models of identity types. *Mathematical Proceedings of the Cambridge Philosophical Society* 146, 1 (2009), 45–55. doi:10.1017/S0305004108001783

[8] J. Baez and M. Stay. 2010. Physics, Topology, Logic and Computation: A Rosetta Stone. In *New Structures for Physics*. Springer, 95–172. arXiv:0903.0340 [quant-ph] doi:10.48550/arxiv.0903.0340

[9] Edwin S. Bainbridge. 1976. Feedback and generalized logic. *Information and Control* 31, 1 (May 1976), 75–96. doi:10.1016/S0019-9958(76)90390-9

[10] Edwin S. Bainbridge, Peter J. Freyd, Andre Scedrov, and Philip J. Scott. 1990. Functorial polymorphism. *Theoretical Computer Science* 70, 1 (Jan. 1990), 35–64. doi:10.1016/0304-3975(90)90151-7

[11] Richard Blute. 1993. Linear logic, coherence and dinaturality. *Theoretical Computer Science* 115, 1 (July 1993), 3–41. doi:10.1016/0304-3975(93)90053-V

[12] R. F. Blute and P. J. Scott. 1996. Linear Läuchli semantics. *Annals of Pure and Applied Logic* 77, 2 (Jan. 1996), 101–142. doi:10.1016/0168-0072(95)00017-8

[13] R. F. Blute and P. J. Scott. 1998. The Shuffle Hopf Algebra and Noncommutative Full Completeness. *Journal of Symbolic Logic* 63, 4 (1998), 1413–1436. doi:10.2307/2586659

[14] Guillaume Boisseau. 2020. String Diagrams for Optics. In *5th International Conference on Formal Structures for Computation and Deduction (FSCD 2020) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 167)*, Zena M. Ariola (Ed.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 17:1–17:18. doi:10.4230/LIPIcs.FSCD.2020.17

[15] Guillaume Boisseau and Jeremy Gibbons. 2018. What you needa know about Yoneda: profunctor optics and the Yoneda lemma (functional pearl). *Proceedings of the ACM on Programming Languages* 2, ICFP (July 2018), 84:1–84:27. doi:10.1145/3236779

[16] Francis Borceux. 1994. *Handbook of Categorical Algebra: Volume 1: Basic Category Theory*. Encyclopedia of Mathematics and its Applications, Vol. 1. Cambridge University Press, Cambridge. doi:10.1017/CBO9780511525858

[17] Claudia Casadio and Philip J. Scott (Eds.). 2021. *Joachim Lambek: The Interplay of Mathematics, Logic, and Linguistics*. Springer Verlag, Cham.

[18] Simon Castellan, Pierre Clairambault, and Peter Dybjer. 2020. *Categories with Families: Unityped, Simply Typed, and Dependently Typed*. Technical Report. arXiv:1904.00827 doi:10.48550/arXiv.1904.00827 arXiv:1904.00827 [cs] type: article.

[19] Fernando Chu, Éléonore Mangel, and Paige Randall North. 2024. A directed type theory for 1-categories. In *30th International Conference on Types for Proofs and Programs TYPES 2024–Abstracts*. 205. https://types2024.itu.dk/abstracts.pdf#page=215

[20] Bryce Clarke, Derek Elkins, Jeremy Gibbons, Fosco Loregian, Bartosz Milewski, Emily Pillmore, and Mario Román. 2022. Profunctor Optics, a Categorical Update. arXiv:2001.07488 [cs.PL] doi:10.48550/arxiv.2001.07488

[21] Jesper Cockx, Nicolas Tabareau, and Théo Winterhalter. 2021. The taming of the rew: a type theory with computational assumptions. *Proceedings of the ACM on Programming Languages* 5, POPL (Jan. 2021), 60:1–60:29. doi:10.1145/3434341

[22] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. 2015. Cubical Type Theory: A Constructive Interpretation of the Univalence Axiom. In *21st International Conference on Types for Proofs and Programs (TYPES 2015) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 69)*, Tarmo Uustalu (Ed.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 5:1–5:34. doi:10.4230/LIPIcs.TYPES.2015.5

[23] Roy L. Crole. 1994. *Categories for Types*. Cambridge University Press, Cambridge. doi:10.1017/CBO9781139172707

[24] G.S.H. Cruttwell and Michael Shulman. 2010. A unified framework for generalized multicategories. *Theory Appl. Categ.* 24 (2010), 580–655. arXiv:0907.2460 doi:10.48550/arxiv.0907.2460

[25] Mario Cáccamo and Glynn Winskel. 2001. A Higher-Order Calculus for Categories. In *Theorem Proving in Higher Order Logics (Lecture Notes in Computer Science)*, Richard J. Boulton and Paul B. Jackson (Eds.). Springer, Berlin, Heidelberg, 136–153. doi:10.1007/3-540-44755-5_11

[26] Eduardo Dubuc and Ross Street. 1970. Dinatural transformations. In *Reports of the Midwest Category Seminar IV (Lecture Notes in Mathematics)*, S. MacLane, H. Applegate, M. Barr, B. Day, E. Dubuc, Phreilambud, A. Pultr, R. Street, M. Tierney, and S. Swierczkowski (Eds.). Springer, Berlin, Heidelberg, 126–137. doi:10.1007/BFb0060443

[27] Samuel Eilenberg and G. M Kelly. 1966. A generalization of the functorial calculus. *Journal of Algebra* 3, 3 (May 1966), 366–375. doi:10.1016/0021-8693(66)90006-8

[28] Lisbeth Fajstrup, Eric Goubault, Emmanuel Haucourt, Samuel Mimram, and Martin Raussen. 2016. *Directed Algebraic Topology and Concurrency*. Springer International Publishing. doi:10.1007/978-3-319-15398-8

[29] Peter J. Freyd, E. P. Robinson, and G. Rosolini. 1992. Dinaturality for free. In *Applications of Categories in Computer Science: Proceedings of the London Mathematical Society Symposium, Durham 1991*, A. M. Pitts, M. P. Fourman, and P. T. Johnstone (Eds.). Cambridge University Press, Cambridge, 107–118. doi:10.1017/CBO9780511525902.007

[30] Peter J. Freyd, Edmund P. Robinson, and Giuseppe Rosolini. 1992. Functorial Parametricity. In *Proceedings of the Seventh Annual Symposium on Logic in Computer Science (LICS '92), Santa Cruz, California, USA, June 22-25, 1992*. IEEE Computer Society, 444–452. doi:10.1109/LICS.1992.185555

[31] Zeinab Galal. 2020. A Profunctorial Scott Semantics. In *5th International Conference on Formal Structures for Computation and Deduction (FSCD 2020) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 167)*, Zena M. Ariola (Ed.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 16:1–16:18. doi:10.4230/LIPIcs.FSCD.2020.16

[32] Jean-Yves Girard, Andre Scedrov, and Philip J. Scott. 1992. Normal Forms and Cut-Free Proofs as Natural Transformations. In *Logic from Computer Science*, Yiannis N. Moschovakis (Ed.). Springer, New York, NY, 217–241. doi:10.1007/978-1-4612-2822-6_8

[33] Marco Grandis. 2009. *Directed Algebraic Topology: Models of Non-Reversible Worlds*. Cambridge University Press.

[34] Daniel Gratzer, Jonathan Weinberger, and Ulrik Buchholtz. 2024. Directed univalence in simplicial homotopy type theory. doi:10.48550/arXiv.2407.09146

[35] Daniel Gratzer, Jonathan Weinberger, and Ulrik Buchholtz. 2025. The Yoneda embedding in simplicial type theory. doi:10.48550/arXiv.2501.13229

[36] Stefano Guerrini. 2004. *Proof Nets and the λ-Calculus*. Cambridge University Press, 65–118.

[37] Robert Harper. 2016. *Practical Foundations for Programming Languages* (2nd ed.). Cambridge University Press, USA.

[38] James Hefford and Cole Comfort. 2023. Coend Optics for Quantum Combs. *Electronic Proceedings in Theoretical Computer Science* 380 (Aug. 2023), 63–76. arXiv:2205.09027 [quant-ph] doi:10.4204/EPTCS.380.4

[39] Chris Heunen and Jamie Vicary. 2019. *Categories for Quantum Theory: An Introduction*. Oxford University Press. arXiv:https://academic.oup.com/book/43710/book-pdf/50991591/9780191060069_web.pdf doi:10.1093/oso/9780198739623.001.0001

[40] Ralf Hinze. 2012. Kan Extensions for Program Optimisation Or: Art and Dan Explain an Old Trick. In *Mathematics of Program Construction (Lecture Notes in Computer Science)*, Jeremy Gibbons and Pablo Nogueira (Eds.). Springer, Berlin, Heidelberg, 324–362. doi:10.1007/978-3-642-31113-0_16

[41] Martin Hofmann. 1997. Syntax and Semantics of Dependent Types. In *Semantics and Logics of Computation*, Andrew M. Pitts and Peter Dybjer (Eds.). Cambridge University Press, Cambridge, 79–130. doi:10.1017/CBO9780511526619.004

[42] Martin Hofmann and Thomas Streicher. 1998. The groupoid interpretation of type theory. In *Twenty-five years of constructive type theory (Venice, 1995)*, Giovanni Sambin and Jan M Smith (Eds.). Oxford Logic Guides, Vol. 36. Oxford Univ. Press, New York, 83–111. doi:10.1093/oso/9780198501275.003.0008

[43] Bart P. F. Jacobs. 1999. *Categorical Logic and Type Theory*. Studies in Logic and the Foundations of Mathematics, Vol. 141. North-Holland.

[44] Alex Kavvos. 2019. A Quantum of Direction. (2019). https://seis.bristol.ac.uk/~tz20861/papers/meio.pdf

[45] Nikolai Kudasov, Emily Riehl, and Jonathan Weinberger. 2024. Formalizing the ∞-Categorical Yoneda Lemma. In *Proceedings of the 13th ACM SIGPLAN International Conference on Certified Programs and Proofs* (London, UK, 2024) *(CPP 2024)*. Association for Computing Machinery, New York, NY, USA, 274–290. doi:10.1145/3636501.3636945

[46] Joachim Lambek and Philip J. Scott. 1986. *Introduction to Higher-Order Categorical Logic*. Cambridge Studies in Advanced Mathematics, Vol. 7. Cambridge University Press. 5https://www.cambridge.org/ee/academic/subjects/mathematics/logic-categories-and-sets/introduction-higher-order-categorical-logic,https://www.cambridge.org/ee/academic/subjects/mathematics/logic-categories-and-sets

[47] F. William Lawvere. 1963. *Functorial Semantics of Algebraic Theories*. Ph. D. Dissertation. Columbia University.

[48] F. William Lawvere. 1969. Adjointness in Foundations. *Dialectica* 23, 3/4 (1969), 281–296. https://www.jstor.org/stable/42969800

[49] F. William Lawvere. 1970. Equality in hyperdoctrines and comprehension schema as an adjoint functor. In *Applications of Categorical Algebra*, A. Heller (Ed.). American Mathematical Society, Providence, R.I., 1–14.
[50] Tom Leinster. 2014. *Basic Category Theory*. Cambridge University Press, Cambridge. doi:10.1017/CBO9781107360068
[51] Daniel R. Licata and Robert Harper. 2011. 2-Dimensional Directed Type Theory. *Electronic Notes in Theoretical Computer Science* 276 (Sept. 2011), 263–289. doi:10.1016/j.entcs.2011.09.026
[52] Fosco Loregian. 2021. *(Co)end Calculus*. Cambridge University Press, Cambridge. doi:10.1017/9781108778657
[53] Saunders Mac Lane. 1998. *Categories for the Working Mathematician* (2nd ed.). Graduate Texts in Mathematics, Vol. 5. Springer-Verlag New York. xii+314 pages. doi:10.1007/978-1-4757-4721-8
[54] Maria Emilia Maietti and Giuseppe Rosolini. 2015. Unifying Exact Completions. *Applied Categorical Structures* 23, 1 (Feb. 2015), 43–52. doi:10.1007/s10485-013-9360-5
[55] Guy McCusker and Alessio Santamaria. 2021. Composing dinatural transformations: Towards a calculus of substitution. *Journal of Pure and Applied Algebra* 225, 10 (Oct. 2021), 106689. doi:10.1016/j.jpaa.2021.106689
[56] Samuel Mimram. 2020. *Program = Proof*. Independently Published. https://books.google.ee/books?id=nzZzzgEACAAJ
[57] Eugenio Moggi. 1991. Notions of computation and monads. *Information and Computation* 93, 1 (1991), 55 – 92. Selections from 1989 IEEE Symposium on Logic in Computer Science.
[58] Jacob Neumann. 2023. *Paranatural Category Theory*. Technical Report. arXiv:2307.09289 doi:10.48550/arXiv.2307.09289 arXiv:2307.09289.
[59] Jacob Neumann. 2025. *A Generalized Algebraic Theory of Directed Equality*. Ph. D. Dissertation.
[60] Max S. New and Daniel R. Licata. 2023. A Formal Logic for Formal Category Theory. In *Foundations of Software Science and Computation Structures (Lecture Notes in Computer Science)*, Orna Kupferman and Pawel Sobocinski (Eds.). Springer Nature Switzerland, 113–134. doi:10.1007/978-3-031-30829-1_6
[61] Paige Randall North. 2019. Towards a Directed Homotopy Type Theory. *Electronic Notes in Theoretical Computer Science* 347 (Nov. 2019), 223–239. doi:10.1016/j.entcs.2019.09.012
[62] Paige Randall North and Fernando Chu. 2025. Dependent two-sided fibrations for directed type theory. (2025).
[63] Andreas Nuyts. 2015. *Towards a Directed Homotopy Type Theory based on 4 Kinds of Variance*. Master's thesis. KU Leuven.
[64] Andreas Nuyts. 2023. Higher Pro-arrows: Towards a Model for Naturality Pretype Theory. (2023).
[65] Robert Paré and Leopoldo Román. 1998. Dinatural numbers. *Journal of Pure and Applied Algebra* 128, 1 (June 1998), 33–92. doi:10.1016/S0022-4049(97)00036-4
[66] Zoran Petrić. 2003. G-dinaturality. *Annals of Pure and Applied Logic* 122, 1 (Aug. 2003), 131–173. doi:10.1016/S0168-0072(03)00003-4
[67] Paolo Pistone. 2017. On Dinaturality, Typability and beta-eta-Stable Models. In *2nd International Conference on Formal Structures for Computation and Deduction (FSCD 2017) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 84)*, Dale Miller (Ed.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 29:1–29:17. doi:10.4230/LIPIcs.FSCD.2017.29
[68] Paolo Pistone. 2018. Proof nets, coends and the Yoneda isomorphism. In *Proceedings Joint International Workshop on Linearity & Trends in Linear Logic and Applications, Linearity-TLLA@FLoC 2018, Oxford, UK, 7-8 July 2018 (EPTCS, Vol. 292)*, Thomas Ehrhard, Maribel Fernández, Valeria de Paiva, and Lorenzo Tortora de Falco (Eds.). 148–167. doi:10.4204/EPTCS.292.9
[69] Paolo Pistone. 2019. On completeness and parametricity in the realizability semantics of System F. *Logical Methods in Computer Science* Volume 15, Issue 4 (Oct. 2019). doi:10.23638/LMCS-15(4:6)2019
[70] Paolo Pistone and Luca Tranchini. 2021. The Yoneda Reduction of Polymorphic Types. In *29th EACSL Annual Conference on Computer Science Logic (CSL 2021) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 183)*, Christel Baier and Jean Goubault-Larrecq (Eds.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 35:1–35:22. doi:10.4230/LIPIcs.CSL.2021.35
[71] Andrew M. Pitts. 1995. Categorical logic. In *Handbook of Logic in Computer Science: Volume 5: Logic and Algebraic Methods*, S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum (Eds.). Oxford University Press, 39–123. doi:10.1093/oso/9780198537816.003.0002
[72] Gordon Plotkin and Martín Abadi. 1993. A logic for parametric polymorphism. In *Typed Lambda Calculi and Applications*, Marc Bezem and Jan Friso Groote (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 361–375.
[73] Emily Riehl and Michael Shulman. 2017. A type theory for synthetic ∞-categories. *Higher structures* 1, 1 (2017). arXiv:1705.07442.
[74] Mario Román. 2020. Open Diagrams via Coend Calculus. In *Electronic Proceedings in Theoretical Computer Science*, Vol. 333. 65–78. arXiv:2004.04526v4 doi:10.4204/EPTCS.333.5
[75] Alessio Santamaria. 2019. *Towards a Godement calculus for dinatural transformations*. Ph. D. Dissertation. University of Bath. https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.787523

[76] P. J. Scott. 2000. Some aspects of categories in computer science. Handbook of Algebra, Vol. 2. North-Holland, 3–77. doi:10.1016/S1570-7954(00)80027-3
[77] Michael Shulman. 2016. *Categorical Logic from a Categorical Point of View.* https://github.com/mikeshulman/catlog
[78] The Univalent Foundations Program. 2013. *Homotopy Type Theory: Univalent Foundations of Mathematics.* https://homotopytypetheory.org/book, Institute for Advanced Study.
[79] Tarmo Uustalu. 2010. A Note on Strong Dinaturality, Initial Algebras and Uniform Parameterized Fixpoint Operators. In *7th Workshop on Fixed Points in Computer Science, FICS 2010, Brno, Czech Republic, August 21-22, 2010*, Luigi Santocanale (Ed.). Laboratoire d'Informatique Fondamentale de Marseille, 77–82. https://hal.archives-ouvertes.fr/hal-00512377/document#page=78
[80] Tarmo Uustalu, Niccolò Veltri, and Noam Zeilberger. 2020. Eilenberg-Kelly Reloaded. *Electronic Notes in Theoretical Computer Science* 352 (Oct. 2020), 233–256. doi:10.1016/j.entcs.2020.09.012
[81] Benno van den Berg and Richard Garner. 2010. Types are weak ω-groupoids. *Proceedings of the London Mathematical Society* 102, 2 (Oct. 2010), 370–394. arXiv:https://academic.oup.com/plms/article-pdf/102/2/370/4487337/pdq026.pdf doi:10.1112/plms/pdq026
[82] Janis Voigtländer. 2020. Free Theorems Simply, via Dinaturality. In *Declarative Programming and Knowledge Management*, Petra Hofstedt, Salvador Abreu, Ulrich John, Herbert Kuchen, and Dietmar Seipel (Eds.). Springer International Publishing, Cham, 247–267. doi:10.1007/978-3-030-46714-2_16
[83] Matthew Weaver. 2024. *Bicubical Directed Type Theory.* Ph. D. Dissertation. https://dataspace.princeton.edu/handle/88435/dsp017s75dg778
[84] Matthew Z. Weaver and Daniel R. Licata. 2020. A Constructive Model of Directed Univalence in Bicubical Sets. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS '20)*. Association for Computing Machinery, New York, NY, USA, 915–928. doi:10.1145/3373718.3394794
[85] R. J. Wood. 1982. Abstract proarrows I. *Cahiers de topologie et géometrie différentielle categoriques* 23, 3 (1982), 279–290.

A Additional judgments for first-order dinatural directed type theory

The rules to formally capture the variance of variables in predicates is given in Figure 14, with the accompanying definition of unused variables in terms in Figure 13.

We show in Figure 15 the full rules in the equational theory regarding cuts. In Figure 16 we explicitly illustrate what a bidirectional rule in "adjoint-form" looks like, by explicitly listing the two directions, the isomorphisms and the naturality conditions.

$$\boxed{\Gamma \ni x : \mathbb{A} \text{ unused in } t : \mathbb{C}} \qquad \frac{\Gamma \ni x : \mathbb{C} \qquad x \neq y}{\Gamma \ni y : \mathbb{C} \text{ unused in } x : \mathbb{C}} \qquad \frac{}{\Gamma \ni x : \mathbb{A} \text{ unused in } ! : \top}$$

$$\frac{\Gamma \ni x : \mathbb{A} \text{ unused in } t : \mathrm{dom}(f)}{\Gamma \ni x : \mathbb{A} \text{ unused in } f(t) : \mathrm{cod}(f)} \qquad \frac{\Gamma \ni x : \mathbb{A} \text{ unused in } t : \mathbb{C}}{\Gamma^{\mathrm{op}} \ni x : \mathbb{A}^{\mathrm{op}} \text{ unused in } t^{\mathrm{op}} : \mathbb{C}^{\mathrm{op}}}$$

$$\frac{\Gamma \ni x : \mathbb{A} \text{ unused in } s : \mathbb{C} \qquad \Gamma \ni x : \mathbb{A} \text{ unused in } t : \mathbb{D}}{\Gamma \ni x : \mathbb{A} \text{ unused in } \langle s, t \rangle : \mathbb{C} \times \mathbb{D}}$$

$$\frac{\Gamma \ni x : \mathbb{A} \text{ unused in } p : \mathbb{C} \times \mathbb{D}}{\Gamma \ni x : \mathbb{A} \text{ unused in } \pi_1(p) : \mathbb{C}} \qquad \frac{\Gamma \ni x : \mathbb{A} \text{ unused in } p : \mathbb{C} \times \mathbb{D}}{\Gamma \ni x : \mathbb{A} \text{ unused in } \pi_2(p) : \mathbb{D}}$$

$$\frac{\Gamma \ni x : \mathbb{A} \text{ unused in } s : [\mathbb{C}, \mathbb{D}] \qquad \Gamma \ni x : \mathbb{A} \text{ unused in } t : \mathbb{C}}{\Gamma \ni x : \mathbb{A} \text{ unused in } s \cdot t : \mathbb{D}} \qquad \frac{\Gamma, x : \mathbb{C} \vdash t(x) : \mathbb{D}}{\Gamma \ni x : \mathbb{A} \text{ unused in } \lambda x.t(x) : [\mathbb{C}, \mathbb{D}]}$$

Fig. 13. Syntax of first-order dinatural directed type theory – syntactically unused variables in terms.

$$\boxed{\Gamma \ni x : \mathbb{A} \text{ cov in } \varphi}$$

$$\frac{\Gamma \ni x : \mathbb{A} \text{ cov in } P \qquad \Gamma \ni x : \mathbb{A} \text{ cov in } Q}{\Gamma \ni x : \mathbb{A} \text{ cov in } P \times Q} \qquad \frac{\Gamma^{\mathrm{op}} \ni x : \mathbb{A}^{\mathrm{op}} \text{ cov in } P \qquad \Gamma \ni x : \mathbb{A} \text{ cov in } Q}{\Gamma \ni x : \mathbb{A} \text{ cov in } P \Rightarrow Q}$$

$$\frac{}{\Gamma \ni x : \mathbb{A} \text{ cov in } \top} \qquad \frac{\Gamma, y : \mathbb{C} \ni x : \mathbb{A} \text{ cov in } \varphi}{\Gamma \ni x : \mathbb{A} \text{ cov in } \int^{y:\mathbb{C}} \varphi(\overline{y}, y)} \qquad \frac{\Gamma, y : \mathbb{C} \ni x : \mathbb{A} \text{ cov in } \varphi}{\Gamma \ni x : \mathbb{A} \text{ cov in } \int_{y:\mathbb{C}} \varphi(\overline{y}, y)}$$

$$\frac{\Gamma^{\mathrm{op}}, \Gamma \ni \overline{x} : \mathbb{A}^{\mathrm{op}} \text{ unused in } s : \mathbb{C}^{\mathrm{op}} \qquad \Gamma^{\mathrm{op}}, \Gamma \ni \overline{x} : \mathbb{A}^{\mathrm{op}} \text{ unused in } t : \mathbb{C}}{\Gamma \ni x : \mathbb{A} \text{ cov in } \hom_{\mathbb{C}}(s, t)}$$

$$\frac{\Gamma^{\mathrm{op}}, \Gamma \ni \overline{x} : \mathbb{A}^{\mathrm{op}} \text{ unused in } s : \mathrm{neg}(P)^{\mathrm{op}} \qquad \Gamma^{\mathrm{op}}, \Gamma \ni \overline{x} : \mathbb{A}^{\mathrm{op}} \text{ unused in } t : \mathrm{pos}(P)}{\Gamma \ni x : \mathbb{A} \text{ cov in } P(s \mid t)}$$

$$\frac{\mathbb{A} = \mathbb{A}' \qquad \varphi = \varphi' \qquad \Gamma \ni x : \mathbb{A} \text{ cov in } \varphi}{\Gamma \ni x : \mathbb{A}' \text{ cov in } \varphi'}$$

$$\boxed{\Gamma \ni x : \mathbb{A} \text{ contra in } \varphi}$$

$$\frac{\Gamma^{\mathrm{op}} \ni x : \mathbb{A}^{\mathrm{op}} \text{ contra in } \varphi^{\mathrm{op}}}{\Gamma \ni x : \mathbb{A} \text{ contra in } \varphi}$$

Fig. 14. Syntax of first-order dinatural directed type theory – syntactic conditions for covariant/contravariant variables in predicates.

$$\boxed{[\Gamma]\ \Phi \vdash \alpha = \beta : P} \quad \cdots$$

$$\frac{\begin{array}{c}\Gamma \text{ unused in } P \text{ and } Q \\ [a : \Delta^{\mathrm{op}}, b : \Delta, x : \Gamma]\ \ \Phi(a, b, \overline{x}, x) \vdash \alpha : P(a, b) \\ [z : \Delta, x : \Gamma]\ k : P(\overline{z}, z), \Phi(\overline{z}, z, \overline{x}, x) \vdash \gamma[k] : Q(\overline{z}, z) \\ [a : \Delta^{\mathrm{op}}, b : \Delta, x : \Gamma]\ k : Q(a, b), \Phi(\overline{b}, \overline{a}, \overline{x}, x) \vdash \beta[k] : R(a, b, \overline{x}, x)\end{array}}{[z : \Delta, x : \Gamma]\ \ \Phi(\overline{z}, z, \overline{x}, x) \vdash (\beta[\gamma])[\alpha] = \beta[\gamma[\alpha]] : R(\overline{z}, z, \overline{x}, x)}\ \text{(assoc-nat-din-nat)}$$

$$\frac{\begin{array}{c}\Gamma \text{ unused in } P, \quad \Delta \text{ unused in } \Phi \\ [a : \Delta]\ k : P(a), \Phi \vdash \alpha[k] : Q(a) \\ [a : \Delta]\ r : Q(a), \Phi \vdash \beta[r] : R(a)\end{array}}{[a : \Delta]\ k : P(a), \Phi \vdash \beta[\alpha]^{\text{cut-nat}} = \beta[\alpha]^{\text{cut-din}} : Q(a)}\ \text{(cut-coherence)}$$

$$\frac{\begin{array}{c}\Gamma \text{ unused in } P \\ [z : \Delta, x : \Gamma]\ k : P(\overline{z}, z), \Phi(\overline{z}, z, \overline{x}, x) \vdash k : P(\overline{z}, z) \\ [a : \Delta^{\mathrm{op}}, b : \Delta, x : \Gamma]\ \ P(a, b), \Phi(\overline{b}, \overline{a}, \overline{x}, x) \vdash \alpha : Q(a, b)\end{array}}{[z : \Delta, x : \Gamma]\ \ P(\overline{z}, z), \Phi(\overline{z}, z, \overline{x}, x) \vdash \alpha[k] = \alpha^{a,b \mapsto z} : Q(\overline{z}, z)}\ (\text{cut-nat-id}_l)$$

$$\frac{\begin{array}{c}\Gamma \text{ unused in } Q \\ [z : \Delta, x : \Gamma]\ \Phi(\overline{z}, z, \overline{x}, x) \vdash \alpha : Q(\overline{z}, z) \\ [a : \Delta^{\mathrm{op}}, b : \Delta, x : \Gamma]\ k : P(a, b), \Phi(\overline{b}, \overline{a}, \overline{x}, x) \vdash k : P(a, b)\end{array}}{[z : \Delta, x : \Gamma]\ \Phi(\overline{z}, z, \overline{x}, x) \vdash k[\alpha] = \alpha : Q(\overline{z}, z)}\ (\text{cut-nat-id}_r)$$

$$\frac{\begin{array}{c}\Gamma \text{ unused in } P \\ [a : \Delta^{\mathrm{op}}, b : \Delta, x : \Gamma]\ k : P(a, b), \Phi(a, b, \overline{x}, x) \vdash k : P(a, b) \\ [z : \Delta, x : \Gamma]\ \ P(\overline{z}, z), \Phi(\overline{z}, z, \overline{x}, x) \vdash \alpha : Q(\overline{z}, z)\end{array}}{[z : \Delta, x : \Gamma]\ \ P(\overline{z}, z), \Phi(\overline{z}, z, \overline{x}, x) \vdash \alpha[k] = \alpha : Q(\overline{z}, z)}\ (\text{cut-din-id}_l)$$

$$\frac{\begin{array}{c}\Gamma \text{ unused in } Q \\ [a : \Delta^{\mathrm{op}}, b : \Delta, x : \Gamma]\ \Phi(a, b, \overline{x}, x) \vdash \alpha : Q(a, b) \\ [z : \Delta, x : \Gamma]\ k : Q(\overline{z}, z), \Phi(\overline{z}, z, \overline{x}, x) \vdash k : Q(\overline{z}, z)\end{array}}{[z : \Delta, x : \Gamma]\ \Phi(\overline{z}, z, \overline{x}, x) \vdash k[\alpha] = \alpha^{a,b \mapsto z} : Q(\overline{z}, z)}\ (\text{cut-din-id}_r)$$

Fig. 15. Syntax of first-order directed type theory – Equational rules for cuts: associativity for natural-dinatural-natural cuts, coherence for cuts between naturals, left and right identities for cut.

$$\boxed{[\Gamma]\ \Phi \vdash \alpha : P} \quad \cdots$$

$$\frac{[x:\mathbb{C},\Gamma]\ \Phi \vdash \alpha : P(\overline{x},x)}{[\Gamma]\ \Phi \vdash \mathsf{end}(\alpha) : \int_{x:\mathbb{C}} P(\overline{x},x)}\ (\text{end}) \qquad \frac{[\Gamma]\ \Phi \vdash \alpha : \int_{x:\mathbb{C}} P(\overline{x},x)}{[x:\mathbb{C},\Gamma]\ \Phi \vdash \mathsf{end}^{-1}(\alpha) : P(\overline{x},x)}\ (\text{end}^{-1})$$

$$\boxed{[\Gamma]\ \Phi \vdash \alpha = \beta : P} \quad \cdots$$

$$\frac{[x:\mathbb{C},\Gamma]\ \Phi \vdash \alpha : P(\overline{x},x)}{[x:\mathbb{C},\Gamma]\ \Phi \vdash \mathsf{end}^{-1}(\mathsf{end}(\alpha)) = \alpha : P(\overline{x},x)} \qquad \frac{[\Gamma]\ \Phi \vdash \alpha : \int_{x:\mathbb{C}} P(\overline{x},x)}{[\Gamma]\ \Phi \vdash \mathsf{end}(\mathsf{end}^{-1}(\alpha)) = \alpha : \int_{x:\mathbb{C}} P(\overline{x},x)}$$

$$\frac{\begin{array}{c}[z:\Delta,\Gamma] \qquad \Phi(\overline{z},z) \vdash \beta : Q(\overline{z},z) \\ [a:\Delta^{\mathrm{op}}, b:\Delta, x:\mathbb{C},\Gamma]\ k : Q(a,b), \Phi(\overline{a},\overline{b}) \vdash \alpha : P(\overline{x},x,a,b)\end{array}}{[x:\mathbb{C}, z:\Delta,\Gamma] \qquad \Phi(\overline{z},z) \vdash \mathsf{end}(\alpha)[\beta] = \mathsf{end}(\alpha[\beta]) : \int_{x:\mathbb{C}} P(\overline{x},x,\overline{z},z)}\ (\text{end-nat}_l)$$

$$\frac{\begin{array}{c}[a:\Delta^{\mathrm{op}}, b:\Delta,\Gamma] \qquad \Phi(a,b) \vdash \beta : Q(a,b) \\ [x:\mathbb{C}, z:\Delta,\Gamma]\ k : Q(\overline{z},z), \Phi(\overline{z},z) \vdash \alpha : P(\overline{x},x,\overline{z},z)\end{array}}{[x:\mathbb{C}, z:\Delta,\Gamma] \qquad \Phi(\overline{z},z) \vdash \mathsf{end}(\alpha)[\beta] = \mathsf{end}(\alpha[\beta]) : \int_{x:\mathbb{C}} P(\overline{x},x,\overline{z},z)}\ (\text{end-din}_l)$$

$$\frac{\begin{array}{c}[x_1:\mathbb{C}^{\mathrm{op}}, x_2:\mathbb{C}, a:\Delta^{\mathrm{op}}, b:\Delta]\ P(x_1,x_2,a,b) \vdash \beta : P'(x_1,x_2,a,b) \\ [x:\mathbb{C}, z:\Delta]\ \Phi(\overline{z},z) \vdash \alpha : P(\overline{x},x,\overline{z},z)\end{array}}{[z:\Delta]\ \Phi(\overline{z},z) \vdash \mathsf{end}_F(\beta)[\mathsf{end}(\alpha)] = \mathsf{end}(\beta[\alpha]) : \int_{x:\mathbb{C}} P'(\overline{x},x,\overline{z},z)}\ (\text{end-din}_r)$$

$$\frac{\begin{array}{c}[x_1:\mathbb{C}^{\mathrm{op}}, x_2:\mathbb{C}, z:\Delta]\ Q(x_1,x_2,\overline{z},z) \vdash \beta : P'(x_1,x_2,\overline{z},z) \\ [x:\mathbb{C}, a:\Delta^{\mathrm{op}}, b:\Delta]\ \Phi(a,b) \vdash \alpha : Q(\overline{x},x,a,b)\end{array}}{[a:\Delta^{\mathrm{op}}, b:\Delta]\ \Phi(a,b) \vdash \mathsf{end}_F(\beta)[\mathsf{end}(\alpha)] = \mathsf{end}(\beta[\alpha]) : \int_{x:\mathbb{C}} P'(\overline{x},x,a,b)}\ (\text{end-nat}_r)$$

Fig. 16. Syntax of first-order directed type theory – Explicit description of a rule in "adjoint-form", e.g., for ends: rules, isomorphisms, and naturality in Φ, P for (end). Naturality in P uses functoriality in Figure 17.

$$\frac{[a:\mathbb{C}^{\mathrm{op}}, b:\mathbb{C},\Gamma]\ k : P(a,b) \vdash \alpha[k] : P(a,b)}{\begin{array}{l}[x:\mathbb{C},\Gamma]\ p : \displaystyle\int_{x:\mathbb{C}} P(\overline{x},x) \vdash \mathsf{end}_F(\alpha) := \mathsf{end}(\alpha[\mathsf{end}^{-1}(p)]) \\ \qquad\qquad\qquad = \mathsf{end}(\alpha)[\mathsf{end}^{-1}(p)] : \displaystyle\int_{x:\mathbb{C}} P(\overline{x},x)\end{array}}$$

Fig. 17. Functoriality of ends for *naturals* by precomposing with the counit of (end).

B Directed type theory, other derivations

Example B.1 (Contractibility of singletons). Recall the derivation for existence of singletons:

$$\frac{}{[\,]\cdot\vdash \mathsf{end}(\mathsf{coend}^{-1}(k)[\mathsf{refl}_x]) : \int_{x:\mathbb{C}^{\mathsf{op}}}\int^{y:\mathbb{C}} \hom_{\mathbb{C}}(x,y)}$$

We now show that singletons are actually contractible: assuming another element $k : \int^{y:\mathbb{C}} \hom_{\mathbb{C}}(x,y)$, we show that it is equal to the the one given in the first derivation (after removing the universal quantifier). Note that the right-hand side must cut away the hypothesis k by precomposing with the constant dinatural !. In the bottom of the derivation we use the fact that the isomorphisms for coends are natural with respect to the cut rules of our type theory. In the top of the derivation we omit for simplicity an application of associativity of cuts and uniqueness of ! which is used to remove the application of J^{-1}.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{[z:\mathbb{C}]\bullet\vdash \mathsf{coend}^{-1}(k)[\mathsf{refl}_z] = \mathsf{coend}^{-1}(k)[\mathsf{refl}_z] : \cdots}\;(\text{refl})}{[z:\mathbb{C}]\bullet\vdash \mathsf{coend}^{-1}(k)[\mathsf{refl}_z] = \mathsf{coend}^{-1}(k)[\mathsf{refl}_z][!][\mathsf{refl}_z] : \cdots}\;(\text{!-unique})+(\text{assoc-nat-din-nat})}{[x:\mathbb{C}^{\mathsf{op}}, y:\mathbb{C}]\; k:\hom_{\mathbb{C}}(x,y)\vdash \mathsf{coend}^{-1}(k) = \mathsf{coend}^{-1}(k)[\mathsf{refl}_x][!] : \cdots}\;(J\text{-eq})}{\cdots = \mathsf{coend}^{-1}(k)[\mathsf{refl}_x][\mathsf{coend}^{-1}(!)] : \cdots}\;(\text{!-unique})}{[x:\mathbb{C}^{\mathsf{op}}, y:\mathbb{C}]\; k:\hom_{\mathbb{C}}(x,y)\vdash \mathsf{coend}^{-1}(s) = \mathsf{coend}^{-1}(\mathsf{coend}^{-1}(k)[\mathsf{refl}_x][!]) : \cdots}\;(\text{coend-natural})}{[x:\mathbb{C}^{\mathsf{op}}]\; k:\int^{y:\mathbb{C}}\hom_{\mathbb{C}}(x,y)\vdash k = \mathsf{coend}^{-1}(k)[\mathsf{refl}_x][!] : \int^{y:\mathbb{C}}\hom_{\mathbb{C}}(x,y)}\;(\text{coend})$$

Example B.2 (Internal naturality for natural transformations). We show that naturality for natural transformations between terms, expressed as ends [52, 1.4.1], holds internally by directed equality elimination. Given terms $\mathbb{C}\vdash F,G:\mathbb{D}$, we use the counit of (end) to extract the family of hom-sets. We first explicitly show the rules used to construct the two sides of a naturality square:

$$\dfrac{\dfrac{\dfrac{[a:\mathbb{C}^{\mathsf{op}}, b:\mathbb{C}]\; f:\hom_{\mathbb{C}}(a,b), \eta:\int_{x:\mathbb{C}}\hom_{\mathbb{D}} F(\overline{x}), G(x)\vdash \eta:\int_{x:\mathbb{C}}\hom(F(\overline{x}),G(x))}{[a:\mathbb{C}^{\mathsf{op}}, b:\mathbb{C}, x:\mathbb{C}]\; f:\hom_{\mathbb{C}}(a,b), \eta:\ldots\vdash \mathsf{end}^{-1}(\eta):\hom(F(\overline{x}),G(x))}\;(\text{end}^{-1})}{[a:\mathbb{C}^{\mathsf{op}}, b:\mathbb{C}]\; f:\hom_{\mathbb{C}}(a,b), \eta:\ldots\vdash \Delta^*(\mathsf{end}^{-1}(\eta)):\hom(F(a),G(\overline{a}))}\;(\text{idx})}{[a:\mathbb{C}^{\mathsf{op}}, b:\mathbb{C}]\; f:\hom_{\mathbb{C}}(a,b), \eta:\ldots\vdash \mathsf{comp}[\Delta^*(\mathsf{end}^{-1}(\eta)), \mathsf{cong}_G[f]]:\hom(F(a),G(b))}\;(\text{cut-nat})$$

where Δ^* is the reindexing functor which collapses a, x to a single variable a, and (cut-nat) is used to apply comp on cong for G. This composition can be done since both cong and comp have the correct naturality shape that allows for (cut-nat) to be applied.

The other derivation is obtained similarly:

$$\dfrac{\dfrac{\dfrac{[a:\mathbb{C}^{\mathsf{op}}, b:\mathbb{C}]\; f:\hom_{\mathbb{C}}(a,b), \eta:\int_{x:\mathbb{C}}\hom_{\mathbb{D}} F(\overline{x}), G(x)\vdash \eta:\int_{x:\mathbb{C}}\hom(F(\overline{x}),G(x))}{[a:\mathbb{C}^{\mathsf{op}}, b:\mathbb{C}, x:\mathbb{C}]\; f:\hom_{\mathbb{C}}(a,b), \eta:\ldots\vdash \mathsf{end}^{-1}(\eta):\hom(F(\overline{x}),G(x))}\;(\text{end}^{-1})}{[a:\mathbb{C}^{\mathsf{op}}, b:\mathbb{C}]\; f:\hom_{\mathbb{C}}(a,b), \eta:\ldots\vdash \Delta^*(\mathsf{end}^{-1}(\eta)):\hom(F(\overline{b}),G(b))}\;(\text{idx})}{[a:\mathbb{C}^{\mathsf{op}}, b:\mathbb{C}]\; f:\hom_{\mathbb{C}}(a,b), \eta:\ldots\vdash \mathsf{comp}[\mathsf{cong}_F[f], \Delta^*(\mathsf{end}^{-1}(\eta))]:\hom(F(a),G(b))}\;(\text{cut-nat})$$

We show that the two maps constructed, corresponding to the two sides of a naturality square, are equal using directed equality elimination; let $K := \Delta^*(\mathsf{end}^{-1}(\eta))$:

$$\dfrac{\dfrac{\dfrac{\overline{[z:\mathbb{C}]\ \ldots \vdash K = K : \mathrm{hom}(F(\overline{z}), G(z))}}{[z:\mathbb{C}]\ \ldots \vdash \mathsf{comp}[\mathsf{refl}_z, K] = \mathsf{comp}[K, \mathsf{refl}_z] : \mathrm{hom}(F(\overline{z}), G(z))}\ (J\text{-comp})}{[z:\mathbb{C}]\ \ldots \vdash \mathsf{comp}[\mathsf{cong}_F[\mathsf{refl}_z], K] = \mathsf{comp}[K, \mathsf{cong}_G[\mathsf{refl}_z]] : \mathrm{hom}(F(\overline{z}), G(z))}\ (J\text{-comp})}{[a:\mathbb{C}^{\mathrm{op}}, b:\mathbb{C}]\ f:\mathrm{hom}_{\mathbb{C}}(a,b), \ldots \vdash \mathsf{comp}[\mathsf{cong}_F[f], K] = \mathsf{comp}[K, \mathsf{cong}_G[f]] : \mathrm{hom}(F(a), G(b))}\ (J\text{-eq})$$

where the equations used follow by the computation rules for cong and left and right unitality of comp. Note that (J-eq) can be used since a, b appear precisely with the correct types that allow for (J) to be applied to contract the equality.

This naturality can then be used to prove a suitable internal Yoneda lemma for the hom of categories by following the standard argument, e.g., given in [50].

Example B.3 (Identity natural transformation). We show the existence of the identity natural transformation for terms, given a functor $\mathbb{C} \vdash F : \mathbb{D}$:

$$\dfrac{\dfrac{}{[x:\mathbb{C}]\ \bullet \vdash F^*(\mathsf{refl}_x) : \mathrm{hom}_{\mathbb{D}}(F(\overline{x}), F(x))}\ (\mathsf{refl})+(\mathsf{idx})}{[\,]\ \bullet \vdash \mathsf{end}^{-1}(l) : \int_{x:\mathbb{C}} \mathrm{hom}_{\mathbb{D}}(F(\overline{x}), F(x))}\ (\mathsf{end})$$

Example B.4 (Composition of natural transformations). We show that natural transformations between terms, expressed as an end [52, 1.4.1], can be composed. Take functors $\mathbb{C} \vdash F, G, H : \mathbb{D}$; first, consider the following elementary derivations:

$$\dfrac{[\,]\ l : \int_{x:\mathbb{C}} \mathrm{hom}_{\mathbb{C}}(F(\overline{x}), G(x)), r : \int_{x:\mathbb{C}} \mathrm{hom}_{\mathbb{C}}(G(\overline{x}), H(x)) \vdash l : \int_{x:\mathbb{C}} \mathrm{hom}_{\mathbb{C}}(F(\overline{x}), G(x))}{[x:\mathbb{C}]\ l : \int_{x:\mathbb{C}} \mathrm{hom}_{\mathbb{C}}(F(\overline{x}), G(x)), r : \int_{x:\mathbb{C}} \mathrm{hom}_{\mathbb{C}}(G(\overline{x}), H(x)) \vdash \mathsf{end}^{-1}(l) : \mathrm{hom}_{\mathbb{C}}(F(\overline{x}), G(x))}\ (\mathsf{end}^{-1})$$

$$\dfrac{[\,]\ l : \int_{x:\mathbb{C}} \mathrm{hom}_{\mathbb{C}}(F(\overline{x}), G(x)), r : \int_{x:\mathbb{C}} \mathrm{hom}_{\mathbb{C}}(G(\overline{x}), H(x)) \vdash l : \int_{x:\mathbb{C}} \mathrm{hom}_{\mathbb{C}}(F(\overline{x}), G(x))}{[x:\mathbb{C}]\ l : \int_{x:\mathbb{C}} \mathrm{hom}_{\mathbb{C}}(F(\overline{x}), G(x)), r : \int_{x:\mathbb{C}} \mathrm{hom}_{\mathbb{C}}(G(\overline{x}), H(x)) \vdash \mathsf{end}^{-1}(r) : \mathrm{hom}_{\mathbb{C}}(G(\overline{x}), H(x))}\ (\mathsf{end}^{-1})$$

Then, we take the statement for transitivity of directed equality, and reindex a with $F(a)$, b with $G(b)$, and c with $H(c)$:

$$\dfrac{\dfrac{}{[a:\mathbb{D}^{\mathrm{op}}, b:\mathbb{D}, c:\mathbb{D}]\ f:\mathrm{hom}_{\mathbb{D}}(a,b),\ g:\mathrm{hom}_{\mathbb{D}}(\overline{b}, c) \vdash \mathsf{comp} : \mathrm{hom}_{\mathbb{D}}(a,c)}\ (J)}{[a:\mathbb{C}^{\mathrm{op}}, b:\mathbb{C}, c:\mathbb{C}]\ f:\mathrm{hom}_{\mathbb{D}}(F(a), G(b)),\ g:\mathrm{hom}(G(\overline{b}), H(c)) \vdash \mathsf{comp}'[f,g] : \mathrm{hom}_{\mathbb{D}}(F(a), H(c))}\ (\mathsf{idx})$$

Now we can perform the composition of this map with the entailments above, which can be done because comp is individually *natural* in a, b, and b, c. Composing l into comp contracts a, b to the same variable z, while still allowing the other map to be later composed in the equality with z, c. Finally, we reintroduce the end quantifier.

$$\dfrac{\dfrac{\dfrac{}{[z:\mathbb{C}^{\mathrm{op}}, c:\mathbb{C}]\ l:\ldots, r:\ldots,\ g:\mathrm{hom}(G(\overline{z}), H(c)) \vdash \mathsf{comp}'[\mathsf{end}^{-1}(l), g] : \mathrm{hom}_{\mathbb{D}}(F(z), H(c))}\ (\mathsf{cut\text{-}nat})}{[w:\mathbb{C}]\ l:\ldots, r:\ldots, \vdash \mathsf{comp}'[\mathsf{end}^{-1}(l), \mathsf{end}^{-1}(r)] : \mathrm{hom}_{\mathbb{D}}(F(\overline{w}), H(w))}\ (\mathsf{cut\text{-}nat})}{[\,]\ l:\ldots, r:\ldots, \vdash \mathsf{end}(\mathsf{comp}'[\mathsf{end}^{-1}(l), \mathsf{end}^{-1}(r)]) : \int_{w:\mathbb{C}} \mathrm{hom}_{\mathbb{D}}(F(\overline{w}), H(w))}\ (\mathsf{end})$$

Associativity of the map above follows from associativity of comp as in the standard case.

Example B.5 (Directed equality in opposite categories). We do not ask that predicates $[x:\mathbb{C}, y:\mathbb{C}^{\mathrm{op}}]\ \mathrm{hom}_{\mathbb{C}^{\mathrm{op}}}(x,y)$ and $[x:\mathbb{C}, y:\mathbb{C}^{\mathrm{op}}]\ \mathrm{hom}_{\mathbb{C}}(y,x)$ are definitionally equal in the equational theory (although this would arguably be a desirable choice), but we can prove by directed equality induction that they are isomorphic:

$$\dfrac{\dfrac{}{[z:\mathbb{C},\Gamma] \qquad \Phi \vdash \mathsf{refl}_z : \hom_{\mathbb{C}}(\overline{z}, z)}\,\text{(refl)}}{[x:\mathbb{C}, y:\mathbb{C}^{\mathrm{op}},\Gamma]\ f:\hom_{\mathbb{C}^{\mathrm{op}}}(x,y), \Phi \vdash J(\mathsf{refl}_z)[f] : \hom_{\mathbb{C}}(y,x)}\,(J)$$

Rule (J) can be applied since x, y appear covariantly in the conclusion. The inverse direction is identical:

$$\dfrac{\dfrac{}{[z:\mathbb{C},\Gamma] \qquad \Phi \vdash \mathsf{refl}_z : \hom_{\mathbb{C}^{\mathrm{op}}}(z, \overline{z})}\,\text{(refl)}}{[x:\mathbb{C}, y:\mathbb{C}^{\mathrm{op}},\Gamma]\ f:\hom_{\mathbb{C}}(y,x), \Phi \vdash J(\mathsf{refl}_z)[f] : \hom_{\mathbb{C}^{\mathrm{op}}}(y,x)}\,(J)$$

In one direction, they compose (since they are both naturals) to the identity by directed equality induction:

$$\dfrac{\dfrac{}{[z:\mathbb{C},\Gamma]\ \Phi \vdash J(\mathsf{refl}_z)[J(\mathsf{refl}_z)[\mathsf{refl}_z]] = J(\mathsf{refl}_z)[\mathsf{refl}_z] = \mathsf{refl}_z : \hom_{\mathbb{C}^{\mathrm{op}}}(z, \overline{z})}\,(J\text{-comp})}{[x:\mathbb{C}, y:\mathbb{C}^{\mathrm{op}},\Gamma]\ f:\hom_{\mathbb{C}}(y,x), \Phi \vdash J(\mathsf{refl}_z)[J(\mathsf{refl}_z)[f]] = f : \hom_{\mathbb{C}^{\mathrm{op}}}(y,x)}\,(J\text{-eq})$$

The other direction is analogous.

C Other rules derivable from the adjoint formulation

The following series of examples captures natural deduction-style rules for coends, where coends are on the right side of the turnstile.

Example C.1 (Elimination for coends). The following derivation captures an elimination rule for coends, where $[\Gamma, d{:}\Delta]\ \Phi(d)$ propctx, $Q(d)$ prop, $[x{:}\mathbb{C}^{\mathrm{op}}, y{:}\mathbb{C}, d{:}\Delta]\ P(x,y,d)$ prop, with variables in Δ always being used *naturally*:

$$\dfrac{\dfrac{\dfrac{[\Gamma, d{:}\Delta]\ \Phi(d) \vdash \int^{x:\mathbb{C}} P(\overline{x},x,d)}{[d{:}\Delta]\ \int^{\gamma:\Gamma} \Phi(\overline{\gamma},\gamma,d) \vdash \int^{x:\mathbb{C}} P(\overline{x},x,d)}\,(\text{coend}^{-1}) \qquad \dfrac{\dfrac{[\Gamma, z:\mathbb{C}, d{:}\Delta]\ P(\overline{z},z,d), \Phi(\overline{\gamma},\gamma,d) \vdash Q(d)}{[\Gamma, d{:}\Delta]\ \int^{z:\mathbb{C}} P(\overline{z},z,d), \Phi(\overline{\gamma},\gamma,d) \vdash Q(d)}\,(\text{coend}^{-1})}{[d{:}\Delta]\ P(\overline{z},z,d), \int^{\gamma:\Gamma} \Phi(\overline{\gamma},\gamma,d) \vdash \int_{\gamma:\Gamma} Q(\overline{\gamma},\gamma,d)}\,(\text{coend}^{-1})+(\text{end})}{[d:\Delta]\ \int^{\gamma:\Gamma} \Phi(\overline{\gamma},\gamma,d) \vdash \int^{\gamma:\Gamma} Q(\overline{\gamma},\gamma,d)}\,(\text{cut-nat})}{[\Gamma, d:\Delta]\ \Phi(d) \vdash Q(d)}\,(\text{coend})+(\text{end}^{-1})$$

Example C.2 (Introduction for coends with a term). The following derivation captures an introduction rule for coends with a generic term $\Delta \vdash F : \mathbb{C}$ (not a *di*term), for $[\Gamma, d:\Delta]\ \Phi(d)$ propctx, $[x:\mathbb{C}, d:\Delta]\ Q(x,d)$ prop:

$$\dfrac{[\Gamma, d:\Delta]\ \Phi(d) \vdash Q(F(d),d) \qquad \dfrac{\dfrac{}{[x:\mathbb{C}, d:\Delta]\ Q(x,d) \vdash \int^{x:\mathbb{C}} Q(x,d)}\,(\text{coend-unit})}{[d:\Delta]\ Q(F(d),d) \vdash \int^{x:\mathbb{C}} Q(x,d)}\,(\text{idx})}{[\Gamma, d:\Delta]\ \Phi(d) \vdash \int^{x:\mathbb{C}} Q(x,d)}$$

In particular, we picked (coend-unit) with Q depending on just a single variable and reindexed with F, which ignores the negative context. Note that variables in Δ are always used naturally.

Example C.3 (Introduction for coends with a dinatural variable). The following derivation captures an introduction rule for coends with a dinatural variable x, for $[x : \mathbb{C}^{\mathrm{op}}, y : \mathbb{C}, \Gamma, d :$

$\Delta]\ \Phi(x,y,d)$ propctx, $[x : \mathbb{C}^{\mathrm{op}}, y : \mathbb{C}, d : \Delta]\ Q(x,y,d)$ prop:

$$\dfrac{[\Gamma, x : \mathbb{C}^{\mathrm{op}}, y : \mathbb{C}]\ \Phi(x,y,d) \vdash Q(x,y,d) \qquad \dfrac{}{[d : \Delta]\ Q(\overline{z}, z, d) \vdash \int^{z:\mathbb{C}} Q(\overline{z}, z, d)}\ \text{(coend-unit)}}{[\Gamma]\ \Phi \vdash \int^{z:\mathbb{C}} Q(\overline{z}, z, d)}\ \text{(cut-din)}$$

In particular, we picked (coend-unit) with Q depending naturally on x, y, z. Note that variables in Δ are always used naturally.

D (Co)end calculus, other derivations

We report here additional examples of derivations for (co)end calculus using our rules.

Example D.1 (Pointwise fomula for left Kan extensions). Dually to Example 6.3, we give a logical proof that the functor $\mathsf{Lan}_F : [\mathbb{C}, \mathsf{Set}] \to [\mathbb{D}, \mathsf{Set}]$ sending (co)presheaves to their left Kan extensions along $F : \mathbb{C} \to \mathbb{D}$ computed via coends [52, 2.3.6] is left adjoint to precomposition $(F;-) : [\mathbb{D}, \mathsf{Set}] \to [\mathbb{C}, \mathsf{Set}]$. For any $[x : \mathbb{C}]\ P(x)$ prop, a functor/term $\mathbb{C} \vdash F : \mathbb{D}$ and a generic $[y : \mathbb{D}]\ \varphi(y)$ prop:

$$\begin{array}{l}
[y : \mathbb{D}] \qquad (\mathsf{Lan}_F P)(x) := \int^{x:\mathbb{C}} \hom_{\mathbb{C}}(F(\overline{x}), y) \times P(x) \vdash \varphi(y) \\
\hline\hline
[x : \mathbb{C}, y : \mathbb{D}]\ \hom_{\mathbb{C}}(F(\overline{x}), y) \times P(x) \vdash \varphi(y) \quad \text{(coend)} \\
\hline\hline
[x : \mathbb{C}, y : \mathbb{D}]\ P(x) \vdash \hom_{\mathbb{C}}(F(x), \overline{y}) \Rightarrow \varphi(y) \quad \text{(exp)} \\
\hline\hline
[x : \mathbb{C}]\ P(x) \vdash \int_{y:\mathbb{D}} \hom_{\mathbb{D}}(F(x), \overline{y}) \Rightarrow \varphi(y) \quad \text{(end)} \\
\hline\hline
[x : \mathbb{C}]\ P(x) \vdash \varphi(F(x)) \quad \text{(Yoneda)}
\end{array}$$

Example D.2 (Right rifts in profunctors). We give a logical proof that composition (on both sides) in Prof has a right adjoint [52, 5.2.5 and Exercise 5.2]. This makes Prof a bicategory where *right extensions* and *right lifts* exist. For simplicity we only treat precomposition, although postcomposition is completely analogous. For any composable profunctors $[x : \mathbb{C}^{\mathrm{op}}, y : \mathbb{A}]\ P(x,y)$ prop, $[x : \mathbb{A}^{\mathrm{op}}, y : \mathbb{D}]\ Q(x,y)$ prop and a generic $[x : \mathbb{C}^{\mathrm{op}}, y : \mathbb{D}]\ \varphi(x,y)$ prop:

$$\begin{array}{l}
[x : \mathbb{C}^{\mathrm{op}}, z : \mathbb{D}] \qquad (P\,;-)(Q)(x,z) := \int^{y:\mathbb{A}} P(x,y) \times Q(\overline{y}, z) \vdash \varphi(x,z) \\
\hline\hline
[x : \mathbb{C}^{\mathrm{op}}, y : \mathbb{A}, z : \mathbb{D}]\ P(x,y) \times Q(\overline{y}, z) \vdash \varphi(x,z) \quad \text{(coend)} \\
\hline\hline
[x : \mathbb{C}^{\mathrm{op}}, y : \mathbb{A}, z : \mathbb{D}]\ Q(\overline{y}, z) \vdash P(\overline{x}, \overline{y}) \Rightarrow \varphi(x,z) \quad \text{(exp)} \\
\hline\hline
[y : \mathbb{A}, z : \mathbb{D}]\ Q(\overline{y}, z) \vdash \int_{x:\mathbb{C}} P(\overline{x}, \overline{y}) \Rightarrow \varphi(x,z) \quad \text{(end)} \\
\hline\hline
[y : \mathbb{A}^{\mathrm{op}}, z : \mathbb{D}]\ Q(y, z) \vdash \int_{x:\mathbb{C}} P(\overline{x}, y) \Rightarrow \varphi(x,z) \quad \text{(op)} \\
\qquad := \mathsf{Rift}_P(\varphi)(y,z)
\end{array}$$

where the last (end) can be applied since $x : \mathbb{C}$ does not appear on the left.

Example D.3 (Composition of profunctors is associative). Using our approach relying on contextual operations we easily show that composition of profunctors, defined via a coend [52], is associative and essentially follows from associativity of products. For composable profunctors $[x : \mathbb{A}^{\mathrm{op}}, y : \mathbb{B}]\ P(x,y)$ prop, $[x : \mathbb{B}^{\mathrm{op}}, y : \mathbb{C}]\ Q(x,y)$ prop, $[x : \mathbb{C}^{\mathrm{op}}, y : \mathbb{D}]\ R(x,y)$ prop, and a generic $[x : \mathbb{A}^{\mathrm{op}}, y : \mathbb{D}]\ \varphi(x,y)$ prop:

$$
\begin{array}{rl}
[a:\mathbb{A}, d:\mathbb{D}]\ \int^{b:\mathbb{B}} P(\overline{a},b)\times\left(\int^{c:\mathbb{C}} Q(\overline{b},c)\times R(\overline{c},d)\right) \vdash \varphi(\overline{a},d) & \\
\hline\hline & \text{(coend)}\\
[a:\mathbb{A}, b:\mathbb{B}, d:\mathbb{D}]\ P(\overline{a},b)\times\left(\int^{c:\mathbb{C}} Q(\overline{b},c)\times R(\overline{c},d)\right) \vdash \varphi(\overline{a},d) & \\
\hline\hline & \text{(coend)}\\
[a:\mathbb{A}, b:\mathbb{B}, c:\mathbb{C}, d:\mathbb{D}]\ P(\overline{a},b)\times(Q(\overline{b},c)\times R(\overline{c},d)) \vdash \varphi(\overline{a},d) & \\
& \text{(structural property)}\\
[a:\mathbb{A}, b:\mathbb{B}, c:\mathbb{C}, d:\mathbb{D}]\ (P(\overline{a},b)\times Q(\overline{b},c))\times R(\overline{c},d) \vdash \varphi(\overline{a},d) & \\
\hline\hline & \text{(coend)}\\
[a:\mathbb{A}, c:\mathbb{C}, d:\mathbb{D}]\ \left(\int^{b:\mathbb{B}} P(\overline{a},b)\times Q(\overline{b},c)\right)\times R(\overline{c},d) \vdash \varphi(\overline{a},d) & \\
\hline\hline & \text{(coend)}\\
[a:\mathbb{A}, d:\mathbb{D}]\ \int^{c:\mathbb{C}}\left(\int^{b:\mathbb{B}} P(\overline{a},b)\times Q(\overline{b},c)\right)\times R(\overline{c},d) \vdash \varphi(\overline{a},d) &
\end{array}
$$

Theorem D.4 (Dinaturals as an end). *The set of dinaturals* $\mathsf{Dinat}(P,Q) := \{P \xrightarrow{\bullet\bullet} Q\}$ *between dipresheaves* $P,Q : \mathbb{C}^{op}\times\mathbb{C} \to \mathsf{Set}$ *can be characterized in terms of the following end [26, Thm. 1],* $\mathsf{Dinat}(P,Q) \cong \int_{x:\mathbb{C}} P(x,\overline{x}) \Rightarrow Q(\overline{x},x)$.

Proof. We give a simple derivation that characterizes all the points (i.e., dinaturals from the point in the empty term context) of the end above using our syntax:

$$
\begin{array}{rl}
\mathsf{Dinat}(P,Q) := [x:\mathbb{C}]\ P(\overline{x},x) \vdash Q(\overline{x},x) & \\
\hline\hline & \text{(exp)}\\
[x:\mathbb{C}]\ \bullet \vdash P(x,\overline{x}) \Rightarrow Q(\overline{x},x) & \\
\hline\hline & \text{(end)}\\
[\,]\ \bullet \vdash \int_{x:\mathbb{C}} P(x,\overline{x}) \Rightarrow Q(\overline{x},x) &
\end{array}
$$

Since dinaturals generalize naturals, a similar derivation justifies the well-known description of natural transformations as ends shown in Section 1 for $F,G:\mathbb{C}\to\mathsf{Set}$,

$$\mathsf{Nat}(F,G) \cong \int_{x:\mathbb{C}} F(\overline{x}) \Rightarrow G(x).$$

□

E Computation rule via J^{-1}

We spell out the proof of the computation rule for the definition of J^{-1} given in Theorem 3.13.

Theorem E.1 ($J^{-1} \iff \mathsf{refl}$). *Rule (refl) is logically equivalent to (J^{-1}); in particular, assuming naturality of J^{-1}, if one defines* $\mathsf{refl}_{\mathbb{C}} := J^{-1}(e)$ *then the computation rule* $J(h)[\mathsf{refl}_{\mathbb{C}}] = h$ *holds in the equational theory.*

Proof. We start by spelling out naturality of J^{-1} in P, which is assumed: explicitly, naturality states that the following two derivations are equal in the equational theory for any α and β (simplifying the context as much as possible for readability):

$$
\dfrac{\dfrac{[a:\mathbb{C}^{op}, b:\mathbb{C}]\ e:\hom_{\mathbb{C}}(a,b), \Phi(\overline{b},\overline{a}) \vdash \alpha : P(a,b)}{[z:\mathbb{C}]\ \Phi(\overline{z},z) \vdash J^{-1}(\alpha[e]) : P(\overline{z},z)} \qquad [z:\mathbb{C}]\ k : P(a,b), \Phi(\overline{a},\overline{b}) \vdash \beta[k] : Q(a,b)}{[z:\mathbb{C}]\ \Phi(\overline{z},z) \vdash \beta[J^{-1}(\alpha)] : Q(\overline{z},z)}
$$

and

$$
\dfrac{\dfrac{[a:\mathbb{C}^{op}, b:\mathbb{C}]\ e:\hom_{\mathbb{C}}(a,b), \Phi(\overline{b},\overline{a}) \vdash \alpha : P(a,b) \qquad [z:\mathbb{C}]\ k : P(a,b), \Phi(\overline{a},\overline{b}) \vdash \beta[k] : Q(a,b)}{[z:\mathbb{C}]\ e:\hom_{\mathbb{C}}(a,b), \Phi(\overline{b},\overline{a}) \vdash \beta[\alpha] : P(\overline{z},z)}}{[z:\mathbb{C}]\ \Phi(\overline{z},z) \vdash J^{-1}(\beta[\alpha]) : Q(\overline{z},z)}
$$

i.e., $\beta[J^{-1}(\alpha)] = J^{-1}(\beta[\alpha])$. In our particular case we take $P(a,b) := \hom(a,b)$ and $\alpha := e$ the projection with (var) and $\beta := J(h)$, from which we obtain that $J(h)[\mathsf{refl}_{\mathbb{C}}] \equiv J(h)[J^{-1}(e)] = J^{-1}(J(h)[e]) = J^{-1}(J(h)) = h$ by the assumption that $J^{-1}(J(h)) = h$ and the fact that (var) is the identity for cut. □

F Frobenius and Beck-Chevalley conditions for (co)ends

Theorem F.1 (Beck-Chevalley and Frobenius condition for (co)ends). *(Co)ends satisfy a* Beck-Chevalley condition, *in the sense that for all* $F : \mathbb{C}^\diamond \to \mathbb{D}$ *there is a strict isomorphism*

$$\int_{\mathbb{A}[\mathbb{D}]} ; F^* \cong (\mathrm{id}_{\mathbb{A}^\diamond} \times F)^* ; \int_{\mathbb{A}[\mathbb{C}]}$$

in the (large) functor category $[[\mathbb{A}^\diamond \times \mathbb{D}^\diamond, \mathrm{Set}], [\mathbb{D}^\diamond, \mathrm{Set}]]$, *where*

$$\int_{\mathbb{A}[\mathbb{C}]}, \int^{\mathbb{A}[\mathbb{C}]} : [\mathbb{A}^\diamond \times \mathbb{C}^\diamond, \mathrm{Set}] \to [\mathbb{C}^\diamond, \mathrm{Set}]$$

are the functors sending dipresheaves to their (co)end in $\mathbb{A}$ *and* $F^* : [\mathbb{D}^\diamond, \mathrm{Set}] \to [\mathbb{C}^\diamond, \mathrm{Set}]$ *is precomposition with* $F^\diamond$.

Moreover, a Frobenius condition *for coends is satisfied, in the sense that there is an isomorphism*

$$\int^{\mathbb{A}[\mathbb{C}]} (\pi^*_{\mathbb{A}[\mathbb{C}]}(P) \times \Phi) \cong \pi^*_{\mathbb{A}[\mathbb{C}]}(P) \times \int^{\mathbb{A}[\mathbb{C}]} (\Phi)$$

natural *in* $\Phi : \mathbb{A}^\diamond \times \mathbb{C}^\diamond \to \mathrm{Set}$, $P : \mathbb{C}^\diamond \to \mathrm{Set}$, *where* $- \times - : [\mathbb{C}, \mathrm{Set}] \times [\mathbb{C}, \mathrm{Set}] \to [\mathbb{C}, \mathrm{Set}]$ *for any* $\mathbb{C}$ *is the product of (di)presheaves.*

Proof. Beck-Chevalley is immediate. For Frobenius, our logical rules can be used to apply exactly the argument given in [43, 1.9.12(i)], detailed in Theorem F.2. □

Theorem F.2 (Frobenius condition for coends). *For any* $\Gamma : \mathbb{A}^\diamond \times \mathbb{C}^\diamond \to \mathrm{Set}$ *and a generic* $K : \mathbb{C}^\diamond \to \mathrm{Set}$, *the following series of derivations gives a logical proof of the Frobenius condition given in Theorem F.1, which we prove by following exactly the argument given in [43, 1.9.12(i)] in the case of fibrations with exponentials. In particular, we show that the Frobenius formulation of (co)ends follows from the non-Frobenius one combined with polarized exponentials. Note that we use the same Yoneda technique described in Remark 6.*

$$[\Gamma] \; \int^{x:\mathbb{A}[\Gamma]} (P \times \Phi(\overline{x}, x)) \vdash \varphi$$
$$\overline{\overline{}} \; \text{(coend-without-frobenius)}$$
$$[x : \mathbb{A}, \Gamma] \; P, \Phi(\overline{x}, x) \vdash \varphi$$
$$\overline{\overline{}} \; \text{(exp)}$$
$$[x : \mathbb{A}, \Gamma] \; \Phi(\overline{x}, x) \vdash P \Rightarrow \varphi$$
$$\overline{\overline{}} \; \text{(coend-without-frobenius)}$$
$$[\Gamma] \; \int^{x:\mathbb{A}[\Gamma]} \Phi(\overline{x}, x) \vdash P \Rightarrow \varphi$$
$$\overline{\overline{}} \; \text{(exp)}$$
$$[\Gamma] \; P, \int^{x:\mathbb{A}[\Gamma]} \Phi(\overline{x}, x) \vdash \varphi$$

Theorem F.3 ((coend-without-frobenius) ⇒ (coend)). *The rule (coend) can be directly justified using (coend-without-frobenius), as follows:*

$$[\Gamma] \left(\int^{a:\mathbb{A}} Q(\overline{a}, a)\right), \Phi \vdash \varphi$$
$$\overline{\overline{}} \; \text{(exp)}$$
$$[\Gamma] \int^{a:\mathbb{A}} Q(\overline{a}, a) \vdash \Phi(x, \overline{x}) \Rightarrow \varphi$$
$$\overline{\overline{}} \; \text{(coend-without-frobenius)}$$
$$[y : \mathbb{C}, \Gamma] \; Q(\overline{a}, a) \vdash \Phi(x, \overline{x}) \Rightarrow \varphi$$
$$\overline{\overline{}} \; \text{(exp)}$$
$$[\Gamma] \; Q(\overline{a}, a), \Phi \vdash \varphi$$

G Yoneda technique

We show how the Yoneda technique described in Remark 6 can be used to prove a derivation of (co)end calculus. We show the case of Yoneda Example 6.1.

$$\begin{array}{c}
[a{:}\mathbb{C}]\ \Phi(a) \vdash \int_{x:\mathbb{C}} \hom_{\mathbb{C}}(a,\overline{x}) \Rightarrow P(x) \\
\hline\hline
[a{:}\mathbb{C}, x{:}\mathbb{C}]\ \Phi(a) \vdash \hom_{\mathbb{C}}(a,\overline{x}) \Rightarrow P(x) \\
\hline\hline
[a{:}\mathbb{C}, x{:}\mathbb{C}]\ \hom_{\mathbb{C}}(\overline{a},x) \times \Phi(a) \vdash P(x) \\
\hline\hline
[z : \mathbb{C}]\ \Phi(z) \vdash P(z)
\end{array}
\begin{array}{l}
\\ (\text{end}) \\ \\ (\text{exp}) \\ \\ (J) \\
\end{array}$$

Explicitly, the two entailments witnessing the isomorphism are obtained by picking Φ to be the context with a single formula and the (var) case at the top of the derivation, i.e.,

$$\dfrac{\dfrac{\dfrac{\dfrac{}{[z : \mathbb{C}]\ k : P(z) \vdash k : P(z)}\ (\text{var})}{[a{:}\mathbb{C}, x{:}\mathbb{C}]\ k : P(a), \hom_{\mathbb{C}}(\overline{a},x) \vdash J(k) : P(x)}\ (J)}{[a{:}\mathbb{C}, x{:}\mathbb{C}]\ k : P(a) \vdash \exp(J(k)) : \hom_{\mathbb{C}}(a,\overline{x}) \Rightarrow P(x)}\ (\text{exp})}{[a{:}\mathbb{C}]\ k : P(a) \vdash \text{end}(\exp(J(k))) : \int_{x:\mathbb{C}} \hom_{\mathbb{C}}(a,\overline{x}) \Rightarrow P(x)}\ (\text{end})$$

and

$$\dfrac{\dfrac{\dfrac{\dfrac{}{[a{:}\mathbb{C}]\ k : \int_{x:\mathbb{C}} \hom_{\mathbb{C}}(a,\overline{x}) \Rightarrow P(x) \vdash k : \int_{x:\mathbb{C}} \hom_{\mathbb{C}}(a,\overline{x}) \Rightarrow P(x)}\ (\text{var})}{[a{:}\mathbb{C}, x{:}\mathbb{C}]\ k : \cdots \vdash \hom_{\mathbb{C}}(a,\overline{x}) \Rightarrow P(x)}\ (\text{end}^{-1})}{[a{:}\mathbb{C}, x{:}\mathbb{C}]\ k : \cdots, \hom_{\mathbb{C}}(\overline{a},x) \vdash P(x)}\ (\exp^{-1})}{[z : \mathbb{C}]\ k : \int_{x:\mathbb{C}} \hom_{\mathbb{C}}(z,\overline{x}) \Rightarrow P(x) \vdash J^{-1}(\exp^{-1}(\text{end}^{-1}(k))) : P(z)}\ (J^{-1})$$

These two entailments can clearly be composed since they are both natural transformations. They compose to the identity in both directions by using the same approach when proving fully faithfulness of the Yoneda embedding [50], i.e., using naturality of each rule in Φ to make them commute with cuts and then using the fact that all rules are invertible:

$$\begin{array}{rl}
\multicolumn{2}{c}{\overline{\hphantom{[a{:}\mathbb{C}]\ k : P(a) \vdash J^{-1}(\exp^{-1}(\text{end}^{-1}(k)))[k \mapsto \text{end}(\exp(J(k)))]}}} \\
\multicolumn{2}{l}{[a{:}\mathbb{C}]\ k : P(a) \vdash J^{-1}(\exp^{-1}(\text{end}^{-1}(k)))[k \mapsto \text{end}(\exp(J(k)))]} \\
& = J^{-1}(\exp^{-1}(\text{end}^{-1}(k))[k \mapsto \text{end}(\exp(J(k)))]) \\
& = J^{-1}(\exp^{-1}(\text{end}^{-1}(k)[k \mapsto \text{end}(\exp(J(k)))])) \\
& = J^{-1}(\exp^{-1}(\text{end}^{-1}(k[k \mapsto \text{end}(\exp(J(k)))]))) \\
& = J^{-1}(\exp^{-1}(\text{end}^{-1}(\text{end}(\exp(J(k)))))) \\
& = J^{-1}(\exp^{-1}(\exp(J(k)))) \\
& = J^{-1}(J(k)) \\
& = k : P(a)
\end{array}$$

Note that we are propagating the cut along the hypothesis k in context (this is only ambiguous in the rule (exp) since there are two hypotheses, where we leave $f : \hom(a, b)$ untouched).

The other direction is obtained analogously.

H Composite in Example 3.8

Given a dinatural transformation

$$[z : \mathbb{C}]\ k : P(\overline{z}, z) \vdash \alpha : Q(\overline{z}, z)$$

we illustrate how the composite

$$[a : \mathbb{C}^{\mathrm{op}}, b : \mathbb{C}]\ f : \hom_{\mathbb{C}}(a, b), k : P(\overline{b}, \overline{a}) \vdash \mathsf{subst}_Q[(f, \mathsf{refl}_a), [\alpha[\mathsf{subst}_P[(\mathsf{refl}_b, f), k]]]] : Q(a, b)$$

in Example 3.8 is indeed allowed by the cut rules of our type theory, i.e., that dinaturals compose. The well-formedness of Example 3.7 follows similarly since it is a special case of the one below. We construct one of the two sides of the equation, with the other one following similarly.

The key idea is that subst is essentially a natural transformation when saturated in the function f (even partially). The subst of a predicate $[a : \mathbb{C}^{\mathrm{op}}, b : \mathbb{C}]\ Q(z, b)$ depending on two variables corresponds to the following entailment:

$$[a', b : \mathbb{C}^{\mathrm{op}}, a, b' : \mathbb{C}]\ f : \hom_{\mathbb{C}}(a', a), g : \hom_{\mathbb{C}}(b, b'), k : Q(\overline{a}, \overline{b}) \vdash \mathsf{subst}_P[f, g, k] : P(a', b')$$

After precomposing f with refl and renaming variables via Theorem 3.14 note that the resulting map is *natural in* z, b after currying the equality g to the right.

$$[b, z : \mathbb{C}^{\mathrm{op}}, b' : \mathbb{C}]\ g : \hom_{\mathbb{C}}(b, b'), k : P(z, \overline{b}) \vdash \mathsf{subst}_P[\mathsf{refl}_z, g, k] : P(z, b')$$

This map *can be precomposed* with α by picking b to be part of the variables of Γ in the rule (cut-din). The intuition for this, described in Section 5 for the semantics of cut, is that one can take the (co)end over b and obtain the above family as *natural* in z and b', without b appearing, which then *can* be composed with α in the expression $\alpha[\mathsf{subst}_P[(\mathsf{refl}_b, f), k]]$. The remaining part of the term is then obtained by using (cut-nat) to compose with subst_Q in an analogous way.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009