Composition of Deductions within the Propositions-As-Types Paradigm

Ivo Pezlar

Abstract. Kosta Došen argued in his papers *Inferential Semantics* ([7]) and *On the Paths of Categories* ([8]) that the propositions-as-types paradigm is less suited for general proof theory because – unlike proof theory based on category theory – it emphasizes categorical proofs over hypothetical inferences. One specific instance of this, Došen points out, is that the Curry-Howard isomorphism makes the associativity of deduction composition invisible. We will show that this is not necessarily the case.

Mathematics Subject Classification (2010). Primary 03F03; Secondary 03B40, 03B15.

Keywords. general proof theory, propositions as types, Curry-Howard isomorphism, constructive type theory, categorial proof theory, cut rule, composition of deduction.

1. Introduction and Motivation

Kosta Došen argued in his papers *Inferential Semantics* ([7]) and *On the Paths of Categories* ([8]) that the propositions-as-types paradigm (see, e.g., [32]) based on the Curry-Howard isomorphism is less suited for general proof theory because – unlike proof theory based on category theory – it makes prominent categorical proofs over hypothetical inferences. One specific instance of this, Došen points out, is that the Curry-Howard isomorphism makes the associativity of deduction composition invisible. This is surprising, because both approaches are known to be equivalent (see [14], [15]). We will examine Došen's claims and argue that approaches based on the Curry-Howard isomorphism do not necessarily favor proofs over inferences and that the associativity of deduction composition does not need to become hidden.

This paper is structured as follows: first, we will briefly introduce notions relevant to our investigation (Section 2), then we will examine Došen's main

Work on this paper was supported by grant no. 19-12420S from the Czech Science Foundation, GA ČR.

argument and offer a counterproposal on how to deal with the composition of deductions within the propositions-as-types paradigm while keeping the associativity visible (Section 3).

2. Preliminary Notes

We start with a short overview of the key vocabulary of this paper:

General proof theory. An approach to proof theory proposed by [26] that views proofs themselves as the primary object of study. Proofs are no longer seen as just a formal device for studying logical consequence but as a subject matter worthy of its own philosophical investigation. See also [13].

Propositions-as-types paradigm. A family of approaches based on the Brouwer-Heyting-Kolmogorov (BHK) interpretation of logical connectives and on the identification of propositions and types, also known as the Curry-Howard isomorphism or correspondence, initially observed by [4] and [10].[1] In its simplest form, it refers to the correspondence between the implicational fragment of intuitionistic natural deduction and typed λ-calculus: rule of assumption, implication introduction (deduction theorem), and implication elimination (modus ponens) on the logical side become, respectively, free variable, abstraction, and application on the functional (computational) side. Propositions are considered collections (types) of their proofs, proving a proposition then becomes equivalent to constructing an object of a certain type, and simplification of proofs corresponds to the evaluation of programs. For example, the implication introduction rule decorated with the corresponding λ-terms (also known as proof objects) will look as follows:

$$\frac{x : A \vdash b(x) : B}{\lambda x.b(x) : A \supset B} \supset\mathrm{I}$$

where $\lambda x.b(x)$ is the abstraction binding the variable x in $b(x)$. In accordance with the propositions-as-types principle, we can read it as either i) if we can construct proof $b(x)$ of proposition B with the assumption that we have a proof x of A, then we can construct a new proof $\lambda x.b(x)$ of $A \supset B$ that depends on no assumptions, or ii) a λ-term $\lambda x.b(x)$ of type $A \supset B$ codes a function that takes an object x of type A and transforms it into an object $b(x)$ of type B (in compliance with the BHK interpretation).

Categorial proof theory. An approach to proof theory based on category theory utilizing the equivalence between typed λ-calculus and Cartesian closed categories found by [14], [15]. The objects of these categories are interpreted as propositions/types and morphisms as terms/proofs. As a specific example, $f : A \to B$ can be understood as stating that the morphism term f codes the deduction from proposition A (premise) to proposition B (conclusion). See also [5], [6]. For example, from the categorial perspective the implication introduction rule will look as follows:

[1] There were, however, other crucial contributors as well, most importantly N. G. de Bruijn and Per Martin-Löf.

$$\frac{h : \top \wedge A \rightarrow B}{h' : \top \rightarrow A \supset B}$$

where $\top$ is a terminal object that behaves like the constant true proposition.

Composition of deductions. The process of splicing together two proofs, where the first ends with a conclusion that is used as an assumption in the other. In other words, it is a cut rule, i.e., a procedure for eliminating auxiliary lemmas from a proof by replacing their occurrences with their proofs. In categorial proof theory, this process is captured via the notion of morphism composition as specified by the following rule (the symbol $\circ$ denotes the binary operation of composition):

$$\frac{f : A \rightarrow B \qquad g : B \rightarrow C}{g \circ f : A \rightarrow C} \ \textsf{CompMorph}$$

read as "if we can deduce B from A and C from B, then C can be directly deduced from A" (see, e.g., [8]). In the propositions-as-types paradigm, this rule can be captured via the notion of substitution as specified by the following rule:

$$\frac{\Gamma \vdash a : A \qquad x : A, \Delta \vdash b : B}{\Gamma, \Delta \vdash b[a/x] : B} \ \textsf{subND}$$

read as "if we can deduce A and also B under the assumption A, then we can substitute the proof of A for the assumption A used in the other proof" (see, e.g., [20]). As a concrete example of deduction composition, consider the following derivation:

$$\frac{\Gamma \vdash A \wedge B \qquad A \wedge B \vdash B}{\Gamma \vdash B}$$

First, we deduced $A \wedge B$ from the assumptions Γ, then we used the same proposition as a further assumption to derive B. Following the general principles of deduction composition, we can see that proposition $A \wedge B$ plays the role of a "grafting" lemma that can be removed in order to join these two proofs. If we do so, we get the deduction B directly from the initial assumptions Γ.

Associativity of deduction composition. This notion captures the idea that the order in which we carry out deduction compositions should not matter, i.e., that we should reach the same conclusion (= permutation of cut). Assuming categorial proof theory in the background, consider, e.g., the following two derivations:

$$\frac{\dfrac{f : A \rightarrow B \qquad g : B \rightarrow C}{g \circ f : A \rightarrow C} \qquad h : C \rightarrow D}{h \circ (g \circ f) : A \rightarrow D}$$

$$\frac{f : A \rightarrow B \qquad \dfrac{g : B \rightarrow C \qquad h : C \rightarrow D}{h \circ g : B \rightarrow D}}{(h \circ g) \circ f : A \rightarrow D}$$

Whether we first compose f with g and then with h or we start by composing g with h and then with f does not affect the outcome, and in both cases we derive the proposition $A \rightarrow D$. Thus we can conclude that associativity holds, i.e., $h \circ (g \circ f) = (h \circ g) \circ f$.

3. Došen's Argument

As mentioned in the previous section, in [7] and [8] it is argued that categorial proof theory is more suited for general proof theory than the propositions-as-types paradigm because it prioritizes inferences (deductions)[2] over categorical proofs.[3] Specifically, Došen writes:

> The typed lambda coding of the Curry-Howard correspondence [...] and the categorial coding in Cartesian closed categories are equivalent in a very precise sense. [...] The import of the two formalisms is however not exactly the same. The typed lambda calculus suggests something different about the subject matter than category theory. It makes prominent the *proofs* $t : B$—and we think immediately of the categorical ones, without hypotheses—while category theory is about the *inferences* $f : A \vdash B$. [It is a] code for a derivation that starts with premise A and ends with conclusion B [and] allow[s] hypotheses to be as visible as conclusions. [The propositions-as-types paradigm] makes conclusions prominent, while hypotheses are veiled. Conclusions are clearly there to be seen as types of terms, while hypotheses are hidden as types of free variables, which are cumbersome to write always explicitly when the variables occur as proper subterms of terms. The desirable terms are closed terms, which code derivations where all the hypotheses have been cancelled. [7]

One of the side-effects of this preference, Došen argues, is that the associativity of deduction composition gets lost:

> [I]n the Curry-Howard correspondence, one designates deductions by typed lambda terms, which is congenial with understanding proofs in the categorical, and not the hypothetical, i.e. categorial, way [...], then composition of deductions is represented by substitution. With that, the associativity of composition becomes invisible, unless one introduces, as it is sometimes done, an explicit substitution operator. [8]

To summarize, Došen raises several concerns about the propositions-as-types paradigm:

1. it is about proofs, not deductions;
2. terms code deductions;
3. desirable terms are closed terms;
4. hypotheses are veiled and cumbersome;
5. associativity of deduction composition is invisible (without an explicit substitution operator).

We will argue against these points and show that adoption of the propositions-as-types paradigm does not commit us to the points above.

3.1. Proofs and Deductions

Došen in [7] states that the Curry-Howard isomorphism emphasizes proofs, while categorial proof theory is about deductions. Although this statement

[2]Došen uses them interchangeably; see e.g., [7], p. 149.

[3]In compliance with the rejection of Schroeder-Heister's first dogma of standard semantics (the priority of categorical over the hypothetical, see [29]).

is not incorrect, it is slightly misleading. Systems built around the Curry-Howard isomorphism are a great deal about deductions as well; they just have a different name for them: *hypothetical judgments*. Take, e.g., constructive/intuitionistic type theory developed by Martin-Löf (CTT; [19]), which can be regarded as a flagship system of the propositions-as-types paradigm. Now, let us consider its rule for implication introduction:[4]

$$\frac{\begin{array}{c} x : A \\ b(x) : B \end{array}}{\lambda x.b(x) : A \supset B} \supset\text{-intro}$$

where the *deduction* premise:[5]

$$\begin{array}{c} x : A \\ b(x) : B \end{array}$$

is nothing other than a hypothetical judgment, i.e., a judgment with a context, that can be also written as $x : A \vdash b(x) : B$. Thus, in CTT, deductions of the form "from A we deduce B" are properly captured as hypothetical judgments, not as lambda terms. The general relationship between $\vdash$ and $\supset$, when A and B are considered as propositions, can be schematized as follows:

$$\underbrace{\underbrace{x : A \vdash b(x) : B}_{\text{hypothetical judgment, sequent, deduction}} \quad \rightsquigarrow \quad \underbrace{\lambda x.b(x) : A \supset B}_{\text{categorical judgment, formula, proof}}}_{\text{“deduction theorem”}}$$

Broadly put, hypothetical judgments carry structural information (similarly to Gentzen's sequents), while the corresponding categorical judgments convey logical information. Furthermore, note that this reduction ("deduction theorem") is not completely faithful. We both gain and lose something from it. What we get is the ability to internalize and express the consequence relation in the form of an implication proposition/type (logical information). What we lose, however, is the dependency feature (structural information).

Remark 3.1. Hypothetical judgments $x : A \vdash b(x) : B$ could also be understood as consequence statements declaring that B follows (proof-theoretically) from A (see [29]) or as function types $A \Rightarrow B$ (see [24]).

Remark 3.2. Hypothetical judgments and conditional judgments (= assertions of implications) should not be conflated. From the CTT perspective, a conditional judgment corresponds to the assertion that a proposition $A \supset B$ is true (i.e., the categorical judgment $a : A \supset B$), which is something different than the assertion that B is true assuming that A is true (i.e., the hypothetical judgment $x : A \vdash b(x) : B$).

---

[4]In CTT, $A \supset B$ is defined via the Π type, i.e., the type of dependent functions, specifically as $(\Pi x : A)B$ where B does not depend on x.

[5]See [25], [23].

3.2. Terms as Codes for Deductions

Došen in [8] claims that within the propositions-as-types paradigm terms code deductions. However, as shown in the previous section, this is imprecise. The most natural way to represent deductions in this paradigm is to use hypothetical judgments of the form $x : A \vdash b(x) : B$, not just lambda terms. Hence, when considering a rule for composing deductions, we should not think of categorical proofs (as Došen probably did, judging by his remarks), but of hypothetical judgments.

3.3. Desirability of Closed Terms

Došen in [7] asserts that within the propositions-as-types paradigm the desirable terms are closed terms, i.e., categorical proofs captured via categorical judgments.

It is true that in systems utilizing this correspondence categorical notions are, conceptually and in terms of explanation, prior to hypothetical ones. For example, in CTT, we start with categorical judgments of the general form $a : A$ (read as "a is an object of type A") and generalize them into hypothetical judgments of the form $x : A \vdash b(x) : B$ (read as "$b(x)$ is an object of type B, assuming x is an object of type A"), i.e., judgments depending on some assumptions, while the meaning of the latter is explained with respect to the former:

> Categorical judgments are conceptually prior to hypothetical judgments [...] It holds in general that the meaning explanation of hypothetical judgments is thus reduced to the meaning explanation of categorical judgments. [27]

However, that does not mean that hypothetical notions are not key in CTT:

> [H]ypothetical judgments are fundamental to the theory. It is, for instance, hypothetical judgments that give rise to the various dependency structures in constructive type theory, by virtue of which it is a dependent type theory. [27]

Thus, although hypothetical judgments are a secondary notion in terms of meaning explanations, by no means are they dispensable or less desirable than categorical judgments. For example, it was dependent types (specifically, dependent function and sum types) introduced with the help of hypothetical judgments that enabled the extension of constructive type theory towards predicate logic.

Remark 3.3. The general attitude towards hypothetical notions is that they are reducible to categorical ones. This approach,[6] popularized by [33] and especially [1], slowly became the standard in both classical and intuitionistic logic and prevails to this day.[7] Of course, not everybody agreed with this

---

[6]It is difficult to surmise who was the first to suggest this reduction, however, it appears as early as the 17th century in the book *Artis Logicae Compendium* by Henry Aldrich (1648–1710) and the general idea was around probably even earlier.
[7]See [29].

endeavour. For example, Frege (1881)[8] in [9] criticized Boole for this and suggested a reverse direction, i.e., reducing categoricals to hypotheticals, or more precisely, reducing Boole's primary propositions into secondary propositions, which included hypotheticals as well.

Remark 3.4. In *The Concept of Mind* Gilbert Ryle wrote:

> Like most dichotomies, the logicians' dichotomy "*either* categorical *or* hypothetical" needs to be taken with a pinch of salt. [...] Save to those who are spellbound by dichotomies, there is nothing scandalous in the notion that a statement may be in some respects like statements of brute fact and in other respects like inference-licences... [28]

Although we do not fully share Ryle's sentiments regarding the nature of the distinction between hypotheticals and categoricals, note that in our case, the "statement of brute [logical] fact" corresponds to $\lambda x.b(x) : A \supset B$, while the "statement of inference-licence" coincides with $x : A \vdash b(x) : B$ since it is a premise of an inference rule.

3.4. Veiled Hypotheses

Došen in [7] argues that within the propositions-as-types paradigm conclusions are prominent, while hypotheses are veiled, i.e., hidden as types of free variables, which are cumbersome to write explicitly. However, as we already demonstrated above, this is also inaccurate. When we represent deductions via hypothetical judgments such as $x : A \vdash b(x) : B$, the conclusion $b(x) : B$ is no more prominent than the hypothesis $x : A$.[9] It is neither veiled nor cumbersome; it is a constitutive part of a hypothetical judgment. Of course, sometimes we might choose to omit contexts to gain more readibility, but that could be hardly counted against the system as such.

3.5. Associativity of Deduction Composition

Došen in [8] states that the associativity of deduction within the propositions-as-types paradigm becomes invisible unless an explicit substitution operator is introduced. We will argue that this is not necessarily the case. First, we reexamine the loss of associativity within the propositions-as-types paradigm and then we propose a way around it.

Suppose we have the following two derivations (3.1) and (3.2) in standard natural deduction, which we want to analyze within the propositions-as-types paradigm while keeping the associativity visible.

$$\dfrac{\dfrac{A \wedge B \vdash A \wedge B \qquad A \wedge B \vdash A}{A \wedge B \vdash A} \qquad A \vdash A \vee B}{A \wedge B \vdash A \vee B} \tag{3.1}$$

$$\dfrac{A \wedge B \vdash A \wedge B \qquad \dfrac{A \wedge B \vdash A \qquad A \vdash A \vee B}{A \wedge B \vdash A \vee B}}{A \wedge B \vdash A \vee B} \tag{3.2}$$

---

[8] Frege's unpublished manuscript *Boole's Logical Calculus and the Concept-Script.*

[9] It is rather the other way around since the conclusion $b(x) : B$ clearly displays its dependence on the variable x from the hypothesis.

Assuming CTT once again as the background theory, we can capture these proofs via the following derivations (3.3) and (3.4) (composing via substitution rule for ND, see above):

$$\dfrac{\dfrac{c : A \wedge B \vdash c : A \wedge B \qquad x : A \wedge B \vdash \mathbf{fst}(x) : A}{c : A \wedge B \vdash \mathbf{fst}(c) : A} \qquad d : A \vdash \mathbf{inl}(d) : A \vee B}{c : A \wedge B \vdash \mathbf{inl}(\mathbf{fst}(c)) : A \vee B} \tag{3.3}$$

$$\dfrac{c : A \wedge B \vdash c : A \wedge B \qquad \dfrac{x : A \wedge B \vdash \mathbf{fst}(x) : A \qquad d : A \vdash \mathbf{inl}(d) : A \vee B}{c : A \wedge B \vdash \mathbf{inl}(\mathbf{fst}(x)) : A \vee B}}{c : A \wedge B \vdash \mathbf{inl}(\mathbf{fst}(c)) : A \vee B} \tag{3.4}$$

Clearly, we have reached the same conclusion despite changing the order of compositions; however, note that the corresponding concluding proof objects $\mathbf{inl}(\mathbf{fst}(c))$ and $\mathbf{inl}(\mathbf{fst}(c))$ do not reflect this since they are identical.

The most straightforward way to record this kind of information is via the notion of associativity. However, associativity is a property of binary operators and, as of now, we have none. Thus, our first task will be to introduce a composition operator for deductions (i.e., hypothetical judgments) in CTT.

As already mentioned above, the categorial representation for deduction $f : A \to B$ corresponds in CTT to the hypothetical judgment $x : A \vdash b(x) : B$ and composition of morphisms $\mathsf{CompMorph}$ roughly corresponds to the substitution of terms for free variables in CTT.[10] With these pieces of information, we can then put together the following rule for deduction composition in CTT (assuming A, B, and C are propositions):

$$\dfrac{x : A \vdash b(x) : B \qquad y : B \vdash c(y) : C}{x : A \vdash c(b(x)) : C}\ \mathsf{CompDed}$$

which can be read as: "assuming that B can be derived from A and that C can be derived from B, then C can be derived from A".

Now, we have a rule for composition, but we are still missing the corresponding composition operator. Let us inspect the way hypothetical judgments are composed in the $\mathsf{CompDed}$ rule, especially the concluding proof object $c(b(x))$ (i.e., the result of the substitution $c[b/y]$). Looking at $c(b(x))$, we should be immediately reminded of the standard function composition, which is defined as a consecutive function application, i.e., $(g \circ f)(x) = g(f(x))$. This similarity is not coincidental, because hypothetical judgments themselves can be considered as functions.[11] Analogously, it seems reasonable to

[10] See [30], [18]. For why we say only "roughly", see e.g., [3].

[11] More specifically, hypothetical judgments can be used to capture a primitive notion of a function which is different from the derived notion of a function captured by the Π type. For more, see [11], [12].

define $(c \circ b)(x)$ simply as $c(b(x))$.[12] Checking whether associativity remains visible is a straightforward task:

$$\dfrac{\dfrac{x : A \vdash b(x) : B \qquad y : B \vdash c(y) : C}{x : A \vdash (c \circ b)(x) : C} \qquad z : C \vdash d(z) : D}{x : A \vdash (d \circ (c \circ b))(x) : D}$$

$$\dfrac{x : A \vdash b(x) : B \qquad \dfrac{y : B \vdash c(y) : C \qquad z : C \vdash d(z) : D}{y : B \vdash (d \circ c)(y) : D}}{x : A \vdash ((d \circ c) \circ b)(x) : D}$$

There is, however, a problem with the CompDed rule and the corresponding definition of composition as we have now presented them. For them to work as intended, the y in $c(y)$ in the second premise of the CompDed rule has to be free, yet we cannot generally guarantee that c indeed contains a free variable. If y does not occur free in c, the concluding proof object $c(b(x))$ becomes just c and the compositionality breaks down. Thus, we need to offer a rule for composition of deductions that does not presuppose the occurrence of a free variable in C.[13] In other words, we need to find a more general way to represent deductions of the general form "from A can be deduced B".

We can achieve this with the higher-order presentation of CTT (see, e.g., [22], [21]) by using the notion of functional abstraction, which allows us to capture and generalize the functional content of hypothetical judgments (deductions) such as $x : A \vdash b : B$. More specifically, assuming A and B are types, we can form a new type $(A)B$, i.e., a type of functions from A to B, which can be populated by the following rule for functional abstraction:[14]

$$\dfrac{x : A \vdash b : B}{(x)b : (A)B}$$

where the prefix notation '( )' indicates the abstraction: all free occurrences of x in b become bound in $(x)b$. Thus, from the perspective of the higher-order presentation of CTT, deductions can be treated as objects of higher-order function types. Changing the rule CompDed accordingly, we get:

$$\dfrac{f : (A)B \qquad g : (B)C}{(g \circ f) : (A)C}\ \textsf{CompDed*}$$

where $(g \circ f)(x) : C$ is defined in a standard manner as $g(f(x)) : C$ in the context $x : A$. Note that this definition no longer presupposes free variables in the proof object g occurring in the second premise in order to be amenable for deduction composition: g itself is an object of a function type $B(C)$ and as such it represents a deduction. This is in contrast to CompDed (and much closer to the rule CompMorph from category theory) where the proof object $c(y)$ occuring in the second premise is an object of a non-functional type

[12]Since it should be always clear from the context, we are overloading the symbol '$\circ$' to mean any kind of composition, i.e., categorial, type-theoretical, or functional.
[13]I am indebted to Ansten Klev, who pointed this out to me and suggested a remedy utilizing the higher-order presentation of CTT presented below.
[14]See [21], p. 143.

C depending on the assumption y of type B and this whole lower-order hypothetical judgment is used to represent a deduction.

To demonstrate the higher-order approach, suppose we have two hypothetical judgments $x : A \vdash b : B$ and $y : B \vdash c : C$. Via the above mentioned abstraction rule, we can construct from them the higher-order functions $(x)b : (A)B$ and $(y)c : (B)C$, respectively, that will constitute the premises for the higher-order deduction composition rule CompDed*. Now, following the rule, we get:

$$((y)c \circ (x)b)(x)$$

which according to the definition of composition can be reduced to:

$$((y)c)(((x)b)(x))$$

which in turn reduces via β-reduction/function application $((x)b)(a) \Rightarrow_\beta b[a/x]$ to:

$$((y)c)(b)$$

and finally to:

$$c[b/y]$$

of type C with the assumption that $x : A$.

Remark 3.5. What kind of object is the $\circ$ operator? In order to properly answer this, we must again utilize the higher-order presentation of CTT, which allows us to type even constants such as $\circ$ that are otherwise unreachable from the lower-order presentation (the same goes, e.g., for Π, λ, $\supset$, $\wedge$). For example, with the higher-order presentation, conjunction $\wedge$ can be defined as an object of type $(\mathit{prop})(\mathit{prop})\mathit{prop}$, i.e., a function that takes two propositions and returns another. Analogously, deduction composition $\circ$ can be defined as an object of type $(A : \mathit{prop})(B : \mathit{prop})(C : \mathit{prop})((A)B)((B)C)(A)C$, i.e., as a function that takes five arguments, specifically propositions A, B, C and functions $(A)B$ and $(B)C$ representing the corresponding deductions (i.e., hypothetical judgments) and returns a function $(A)C$.

Now, once we are equipped with the proper rule and composition operator, let us analyze our earlier derivations (3.1) and (3.2). We get derivations (3.5) and (3.6):

$$\dfrac{\dfrac{f : (A \wedge B)A \wedge B \qquad \mathbf{fst} : (A \wedge B)A}{\mathbf{fst} \circ f : (A \wedge B)A} \qquad \mathbf{inl} : (A)A \vee B}{\mathbf{inl} \circ (\mathbf{fst} \circ f) : (A \wedge B)A \vee B} \tag{3.5}$$

$$\dfrac{f : (A \wedge B)A \wedge B \qquad \dfrac{\mathbf{fst} : (A \wedge B)A \qquad \mathbf{inl} : (A)A \vee B}{\mathbf{inl} \circ \mathbf{fst} : (A \wedge B)A \vee B}}{(\mathbf{inl} \circ \mathbf{fst}) \circ f : (A \wedge B)A \vee B} \tag{3.6}$$

With associativity present, we can see that the different order of compositions is now finally reflected at the level of proof objects as well: for different permutations of cut we have different yet equivalent proof objects $(\mathbf{inl} \circ (\mathbf{fst} \circ f)) = ((\mathbf{inl} \circ \mathbf{fst}) \circ f)$.

4. Conclusion

In this paper, we have tried to show that, contrary to Došen's claims, the propositions-as-types paradigm does not favour categorical proofs over inferences and that the associativity of deduction composition does not have to become invisible. We have demonstrated this in CTT, where deductions are understood in terms of hypothetical judgments. Hypothetical judgments can be composed while keeping track of their associativity (with the help of higher-order presentation of CTT) and they also meet most of Došen's desiderata for proper representations of inferences [7]: their assumptions are not veiled or hidden as types of free variables, they are not cumbersome to write always explicitly and they are just as prominent as conclusions. And although we have chosen here CTT as the representative of a Curry-Howard isomorphism-based framework, our general points can be applied to other related systems containing hypothetical judgments as well (e.g., calculus of constructions by [2], unifying theory of dependent types by [16], [17] or homotopy type theory as presented in [31]). Of course, it still might turn out in the end that categorial proof theory is a more suitable framework for proof analysis than the propositions-as-types paradigm, but, based on the conclusions we have reached here, we cannot agree with Došen's reasons for postulating the superiority of the former.

Acknowledgments

An earlier version of this paper was presented at a seminar organized by the Department of Logic of the Institute of Philosophy (The Czech Academy of Sciences) in Prague, February 2019. I would like to thank all the participants of this seminar for their helpful notes. A special thanks goes to Ansten Klev, whose valuable remarks helped to shape this paper.

References

[1] George Boole. *An Investigation of the Laws of Thought*. Walton & Maberly, 1854.

[2] Thierry Coquand and Gerard Huet. The calculus of constructions. *Information and Computation*, 76(2-3):95–120, 2 1988.

[3] Pierre-Louis Curien, Richard Garner, and Martin Hofmann. Revisiting the categorical interpretation of dependent type theory. *Theoretical Computer Science*, 546:99–119, 8 2014.

[4] H B Curry and R Feys. *Combinatory Logic*, volume 1 of *Combinatory Logic*. North-Holland Publishing Company, 1958.

[5] Kosta Došen. Deductive Completeness. *Bulletin of Symbolic Logic*, 2(03):243–283, 9 1996.

[6] Kosta Došen. Abstraction and application in adjunction. In Z. Kadelburg, editor, *Proceedings of the Tenth Congress of Yugoslav Mathematicians, Faculty of Mathematics*, pages 33–46, Belgrade, 2001. University of Belgrade.

[7] Kosta Došen. Inferential Semantics. In Heinrich Wansing, editor, *Dag Prawitz on Proofs and Meaning*, pages 147–162. Springer International Publishing, Switzerland, 2015.

[8] Kosta Došen. On the Paths of Categories. In Thomas Piecha and Peter Schroeder-Heister, editors, *Advances in Proof-Theoretic Semantics*, pages 65–77. Springer International Publishing, Cham, 2016.

[9] Gottlob Frege. *Posthumous Writings*. Wiley, 1979.

[10] William Alvin Howard. The formulae-as-types notion of construction. In Haskell B. Curry, J. Roger Hindley, and Jonathan P. Seldin, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*. Academic Press, 1980.

[11] Ansten Klev. A comparison of type theory with set theory. In Stefania Centrone, Deborah Kant, and Deniz Sarikaya, editors, *Reflections on the Foundations of Mathematics*. Springer, Cham, 2019.

[12] Ansten Klev. Name of the Sinus Function. In Igor Sedlár and Martin Blicha, editors, *The Logica Yearbook 2018*, London, 2019. College Publications.

[13] Georg Kreisel. A Survey of Proof Theory II. In J. E. Renstad, editor, *Proceedings of the Second Scandinavian Logic Symposium,*, page 109–170, Amsterdam, 1971. North-Holland.

[14] Joachim Lambek. Functional completeness of cartesian categories. *Annals of Mathematical Logic*, 6(3-4):259–292, 3 1974.

[15] Joachim Lambek and P. J. Scott. *Introduction to higher order categorical logic*. Cambridge University Press, 1986.

[16] Zhaohui Luo. *An Extended Calculus of Constructions*. PhD thesis, University of Edinburgh, 1990.

[17] Zhaohui Luo. *Computation and reasoning: a type theory for computer science*. Clarendon Press, 1994.

[18] Per Martin-Löf. An Intuitionistic Theory of Types: Predicative Part. In H E Rose and J C Shepherdson, editors, *Logic Colloquium '73 Proceedings of the Logic Colloquium*, volume 80 of *Studies in Logic and the Foundations of Mathematics*, pages 73–118. Elsevier, 1975.

[19] Per Martin-Löf. *Intuitionistic type theory*. Studies in proof theory. Bibliopolis, 1984.

[20] Sara Negri, Jan von Plato, and Aarne Ranta. *Structural Proof Theory*. Cambridge University Press, 2001.

[21] Bengt Nordström, Kent Petersson, and Jan M Smith. *Programming in Martin-Löf's type theory: an introduction*. International series of monographs on computer science. Clarendon Press, 1990.

[22] Bengt Nordström, Kent Petersson, and Jan M Smith. *Martin-Löf's type theory, Handbook of logic in computer science: Volume 5: Logic and algebraic methods*. Oxford University Press, Oxford, 2001.

[23] Ivo Pezlar. Towards a More General Concept of Inference. *Logica Universalis*, 8(1), 2014.

[24] Ivo Pezlar. The Placeholder View of Assumptions and the Curry–Howard Correspondence. *Synthese*, 2020.

[25] Dag Prawitz. *Natural Deduction: A Proof-theoretical Study*. Dover Books on Mathematics Series. Dover Publications, Incorporated, 1965.

[26] Dag Prawitz. The Philosophical Position of Proof Theory. In R. E. Olson and A. M. Paul, editors, *Contemporary Philosophy in Scandinavia*, page 123–134. John Hopkins Press, Baltimore, 1972.

[27] Shahid Rahman, Zoe McConaughey, Ansten Klev, and Nicolas Clerbout. A Brief Introduction to Constructive Type Theory. In *Immanent Reasoning or Equality in Action*, pages 17–55. Springer, Cham, 2018.

[28] Gilbert Ryle. *The Concept of Mind*. University of Chicago Press, 1949.

[29] Peter Schroeder-Heister. The categorical and the hypothetical: a critique of some fundamental assumptions of standard semantics. *Synthese*, 187(3):925–942, 8 2012.

[30] R. A. G. Seely. Locally cartesian closed categories and type theory. *Mathematical Proceedings of the Cambridge Philosophical Society*, 95(01):33, 1 1984.

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

[32] Philip Wadler. Propositions as Types. *Communications of the ACM*, 58(12):75–84, 2015.

[33] Richard Whatley. *Elements of Logic*. Printed for J. Mawman, London, 2nd editio edition, 1827.

Ivo Pezlar
The Czech Academy of Sciences
Institute of Philosophy
Jilská 1
110 00 Prague 1
Czech Republic
e-mail: `pezlar@flu.cas.cz`