Associativity of deduction composition in natural deduction

Ivo Pezlar

Czech Academy of Sciences, Institute of Philosophy
Prague, Czech Republic

April 2022, online

(Logic4Peace)
Table of Contents

1  Introduction and Motivation
2  Preliminary Notes
3  Composition of Deductions
4  Conclusion
Introduction and Motivation

Outline

1. Introduction and Motivation
2. Preliminary Notes
3. Composition of Deductions
4. Conclusion
Introduction and Motivation

Aim

► Kosta Došen argued in his papers *Inferential Semantics* (2015) and *On the Paths of Categories* (2016) that the propositions as types paradigm is less suited for general proof theory because – unlike categorial proof theory based – 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. I will argue that this is not necessarily the case [Pezlar, 2020].
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* $f : A \vdash B$. [Došen, 2015]
In 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. [Došen, 2016]
Outline

1. Introduction and Motivation
2. Preliminary Notes
3. Composition of Deductions
4. Conclusion
Preliminary Notes

- general proof theory
- propositions as types principle
- categorial proof theory
- composition of deductions
- associativity
General proof theory

- general proof theory (vs. reductive theory):

  ... proofs are studied in their own right in the hope of understanding their nature ... [Prawitz, 1972]

  Proofs and their representations by formal derivations are treated as principal objects of study, not as mere tools for analyzing the consequence relation. [Kreisel, 1971]
Propositions as types

▶ see [Curry and Feys, 1958], [Howard, 1980], [De Bruijn, 1968]
▶ a proposition as the collection (type) of its proofs
▶ proving a proposition as inhabiting a type
▶ proofs as programs
▶ simplification of proofs as evaluation of programs

Example:

$$
\frac{[A \land B]_1}{(A \land B) \rightarrow A} \xrightarrow{\land E_L} \lambda x.fst(x) : (A \land B) \rightarrow A
$$

$$
\frac{x : [A \land B]_1}{fst(x) : A}
$$
Propositions as types

▶ see [Curry and Feys, 1958], [Howard, 1980], [De Bruijn, 1968]
▶ a proposition as the collection (type) of its proofs
▶ proving a proposition as inhabiting a type
▶ proofs as programs
▶ simplification of proofs as evaluation of programs

Example:

\[
\begin{align*}
&\frac{}{[A \land B]_1} \\
&\frac{A}{(A \land B) \rightarrow A} \quad \rightarrow^1 \\
&\frac{\land E_L}{A} \\
&\frac{x : [A \land B]_1}{\text{fst}(x) : A} \\
&\frac{\lambda x.\text{fst}(x) : (A \land B) \rightarrow A}{\lambda x.\text{fst}(x) : (A \land B) \rightarrow A}
\end{align*}
\]
Categorial proof theory

▶ see [Došen, 1996], [Došen, 2001], [Lambek, 1974], [Lambek and Scott, 1986]
▶ Curry-Howard-Lambek correspondence
▶ objects interpreted as types/propositions and arrows as terms/proofs
▶ \( f : A \vdash B \) as a code for a deduction that starts with premise \( A \) and ends with conclusion \( B \)
Composition of deductions

▶ categorial proof theory: composition of deductions = composition of arrows

\[
\begin{align*}
  f : A \rightarrow B \\
g : B \rightarrow C \\
g \circ f : A \rightarrow C
\end{align*}
\]

ArrComp

▶ proposition as types: composition of deductions = substitution

\[
\begin{align*}
  \Gamma \vdash a : A \\
x : A, \Delta \vdash b : B \\
\Gamma, \Delta \vdash b[a/x] : B
\end{align*}
\]

subs-ND

▶ Example:

\[
\begin{align*}
  \Gamma \vdash A \land B \\
  A \land B \vdash B \\
  \Gamma \vdash B
\end{align*}
\]
Preliminary Notes

Composition of deductions

- categorial proof theory: composition of deductions = composition of arrows
  \[
  f : A \rightarrow B \quad g : B \rightarrow C
  \]
  \[
  \frac{}{g \circ f : A \rightarrow C}
  \]
  ArrComp

- proposition as types: composition of deductions = substitution
  \[
  \frac{\Gamma \vdash a : A \quad x : A, \Delta \vdash b : B}{\Gamma, \Delta \vdash b[a/x] : B}
  \]
  subs-ND

- Example:
  \[
  \frac{\Gamma \vdash A \land B \quad A \land B \vdash B}{\Gamma \vdash B}
  \]
Composition of deductions

- Categorial proof theory: composition of deductions = composition of arrows

\[
\frac{f : A \to B \quad g : B \to C}{g \circ f : A \to C} \quad \text{ArrComp}
\]

- Proposition as types: composition of deductions = substitution

\[
\frac{\Gamma \vdash a : A \quad x : A, \Delta \vdash b : B}{\Gamma, \Delta \vdash b[a/x] : B} \quad \text{subs-ND}
\]

- Example:

\[
\frac{\Gamma \vdash A \land B \quad A \land B \vdash B}{\Gamma \vdash B}
\]
Associativity of deductions

- Associativity of deduction composition = permutation of cut

\[ \vdots \text{the binary operation of composition [...] which in terms of deductions is a simple form of cut of sequent systems.} \vdots \text{[Došen, 2016]} \]

\[
f : A \to B \quad g : B \to C \quad h : C \to D \\
g \circ f : A \to C \quad h : C \to D \\
h \circ (g \circ f) : A \to D \\
\]

\[
g : B \to C \quad h : C \to D \\
h \circ g : B \to D \\
(h \circ g) \circ f : A \to D \\
\]

\[
h \circ (g \circ f) = (h \circ g) \circ f
\]
Preliminary Notes

Associativity of deductions

▶ Associativity of deduction composition = permutation of cut

... the binary operation of composition [...] which in terms of deductions is a simple form of cut of sequent systems... [Došen, 2016]

\[
f : A \rightarrow B \quad g : B \rightarrow C
\]

\[
g \circ f : A \rightarrow C \quad h : C \rightarrow D
\]

\[
h \circ (g \circ f) : A \rightarrow D
\]

\[
g : B \rightarrow C \quad h : C \rightarrow D
\]

\[
f : A \rightarrow B \quad h \circ g : B \rightarrow D
\]

\[
(h \circ g) \circ f : A \rightarrow D
\]

\[
h \circ (g \circ f) = (h \circ g) \circ f
\]
Associativity of deductions

▶ Associativity of deduction composition = permutation of cut

... the binary operation of composition [...] which in terms of deductions is a simple form of cut of sequent systems... [Došen, 2016]

\[
\begin{align*}
  f : A \rightarrow B & \quad g : B \rightarrow C \\
  g \circ f : A \rightarrow C & \quad h : C \rightarrow D \\
  h \circ (g \circ f) : A \rightarrow D \\
  g \circ B \rightarrow C & \quad h : C \rightarrow D \\
  f : A \rightarrow B & \quad (h \circ g) : B \rightarrow D \\
  (h \circ g) \circ f : A \rightarrow D \\
  h \circ (g \circ f) = (h \circ g) \circ f
\end{align*}
\]
Composition of Deductions

Outline

1. Introduction and Motivation
2. Preliminary Notes
3. Composition of Deductions
4. Conclusion
Došen: “Curry-Howard correspondence makes prominent proofs, while categorial proof theory is about deductions”

systems built around the propositions as types principle, such as, e.g., constructive type theory ([Martin-Löf, 1984], CTT), are about deductions as well, they just have a different name for them: hypothetical judgments
Composition of Deductions

Deductions: 2/4

In CTT, we start with categorical judgments

\[ a : A \]

and generalize them into hypothetical judgments

\[ x : A \vdash b(x) : B(x) \]

i.e., judgments depending on some assumptions, while the meaning of the latter is explained w.r.t. the former

However, that does not mean that hypothetical notions are dispensable in CTT
Composition of Deductions

Deductions: 2/4

▶ In CTT, we start with categorical judgments

\[ a : A \]

and generalize them into hypothetical judgments

\[ x : A \vdash b(x) : B(x) \]

i.e., judgments depending on some assumptions, while the meaning of the latter is explained w.r.t. the former

▶ However, that does not mean that hypothetical notions are dispensable in CTT
Consider, e.g., the rule for implication introduction:

\[
\frac{x : A \\
b(x) : B}{\lambda x. b(x) : A \to B} \rightarrow \text{intro}
\]

where the *deduction* premise:

\[
x : A \\
b(x) : B
\]

is nothing other than a *hypothetical judgment*, i.e., a judgment with a context, that can be also written as \( x : A \vdash b(x) : B \)
Composition of Deductions

Deductions: 4/4

The relationship between $\vdash$ and $\rightarrow$, when $A$ and $B$ are considered as propositions, can be schematized as follows:

\[ x : A \vdash b(x) : B \rightarrow \lambda x. b(x) : A \rightarrow B \]

hypothetical judgment, sequent, deduction  categorical judgment, formula, proof

“deduction theorem”

▶ structural vs. logical information
▶ when considering a rule for composing deductions, we should think of hypothetical judgments and not of categorical ones
The relationship between $\vdash$ and $\rightarrow$, when $A$ and $B$ are considered as propositions, can be schematized as follows:

\[
\frac{x : A \vdash b(x) : B}{\Rightarrow} \quad \frac{\lambda x.b(x) : A \rightarrow B}{\text{hypothetical judgment, sequent, deduction}} \quad \text{categorical judgment, formula, proof}
\]

“deduction theorem”

- structural vs. logical information
- when considering a rule for composing deductions, we should think of hypothetical judgments and not of categorical ones
Composition of Deductions

Composing deductions: 1/8

\[
\begin{align*}
A \land B &\vdash A \land B \\
\hline
A \land B &\vdash A
\end{align*}
\]

\[
A \land B \vdash A
\]

\[
\begin{align*}
A \land B &\vdash A \land B \\
\hline
A \vdash A \lor B
\end{align*}
\]

\[
A \land B \vdash A \lor B
\]

(1)

\[
\begin{align*}
A \land B &\vdash A \\
A \vdash A \lor B
\end{align*}
\]

\[
\begin{align*}
A \land B &\vdash A \land B \\
\hline
A \land B &\vdash A \lor B
\end{align*}
\]

(2)
Composition of Deductions

Composing deductions: 2/8

\[
\begin{align*}
  c : A \land B \vdash c : A \land B & \quad x : A \land B \vdash \text{fst}(x) : A \\
  c : A \land B \vdash \text{fst}(c) : A & \quad d : A \vdash \text{inl}(d) : A \lor B \\
  c : A \land B \vdash \text{inl}(\text{fst}(c)) : A \lor B \\
  x : A \land B \vdash \text{fst}(x) : A & \quad d : A \vdash \text{inl}(d) : A \lor B \\
  c : A \land B \vdash c : A \land B & \quad c : A \land B \vdash \text{inl}(\text{fst}(x)) : A \lor B \\
  c : A \land B \vdash \text{inl}(\text{fst}(c)) : A \lor B
\end{align*}
\]

Clearly $\text{inl}(\text{fst}(c)) = \text{inl}(\text{fst}(c)) : A \lor B$, but no associativity.
Composition of Deductions

Composing deductions: 2/8

\[
\begin{array}{c}
c : A \land B \vdash c : A \land B \\
x : A \land B \vdash \text{fst}(x) : A \\
c : A \land B \vdash \text{fst}(c) : A \\
d : A \vdash \text{inl}(d) : A \lor B \\
c : A \land B \vdash \text{inl}(\text{fst}(c)) : A \lor B \\
\end{array}
\]

\[
\begin{array}{c}
x : A \land B \vdash \text{fst}(x) : A \\
d : A \vdash \text{inl}(d) : A \lor B \\
c : A \land B \vdash \text{inl}(\text{fst}(x)) : A \lor B \\
\end{array}
\]

Clearly \( \text{inl}(\text{fst}(c)) = \text{inl}(\text{fst}(c)) : A \lor B \), but no associativity
Composition of Deductions

Composing deductions: 3/8

Composition of arrows in category theory corresponds to substitution in constructive type theory: the arrows are interpreted as terms, objects as types:

\[
\frac{x : A \vdash b(x) : B \quad y : B \vdash c(y) : C}{x : A \vdash c(b(x)) : C} \quad \text{CompDed}
\]

We can rewrite \(c(b(x))\) as \((c \circ b)(x)\)
Composition of Deductions

Composing deductions: 4/8

\[
\begin{align*}
  x : A &\vdash b(x) : B \\
  y : B &\vdash c(y) : C
  \end{align*}
\]

\[
\begin{array}{c}
  x : A \vdash (c \circ b)(x) : C \\
  z : C \vdash d(z) : D
\end{array}
\]

\[
\begin{align*}
  x : A &\vdash (d \circ (c \circ b))(x) : D
  \end{align*}
\]
Composition of Deductions

Composing deductions: 5/8

- There is, however, a problem with the CompDed rule as presented.
- The $y$ in $c(y)$ in the second premise of the CompDed rule has to be free, otherwise the compositionality breaks down.
- Yet we cannot generally guarantee that $c$ contains a free variable.
- 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., [Nordström et al., 2001]) by using the notion of functional abstraction

it allows us to express and generalize the functional content of hypothetical judgments (deductions) such as $x : A \vdash b : B$

assuming $A$ and $B$ are types, we can form a new type $(A)B$, which can be populated by the following rule for functional abstraction:

\[
\frac{x : A \vdash b : B}{(x)b : (A)B}
\]
Composition of Deductions

Composing deductions: 7/8

- deductions can be treated as objects of higher-order function types. Changing the rule CompDed accordingly, we get:

\[
\begin{align*}
  f : (A)B & \quad g : (B)C \\
  (g \circ f) & : (A)C \\
\end{align*}
\]

CompDed*

where \((g \circ f)(x) : C\) is defined in a standard manner as \(g(f(x)) : C\) in the context \(x : A\).
Composition of Deductions

Composing deductions: 8/8

\[ f : (A \land B)A \land B \quad \text{fst} : (A \land B)A \]
\[ \text{fst} \circ f : (A \land B)A \quad \text{inl} : (A)A \lor B \]
\[ \text{inl} \circ (\text{fst} \circ f) : (A \land B)A \lor B \quad (3) \]

\[ f : (A \land B)A \land B \quad \text{fst} : (A \land B)A \quad \text{inl} : (A)A \lor B \]
\[ \text{inl} \circ \text{fst} : (A \land B)A \lor B \]
\[ ((\text{inl} \circ \text{fst}) \circ f) : (A \land B)A \lor B \quad (4) \]

For different permutations of cut we have different yet equivalent proof objects: \((\text{inl} \circ (\text{fst} \circ f)) = ((\text{inl} \circ \text{fst}) \circ f)\).
Composition of Deductions

Composing deductions: 8/8

\[
\begin{align*}
  f &: (A \land B)A \land B & \quad \text{fst} &: (A \land B)A \\
  \text{fst} \circ f &: (A \land B)A & \quad \text{inl} &: (A)A \lor B \\
  \text{inl} \circ (\text{fst} \circ f) &: (A \land B)A \lor B
\end{align*}
\] (3)

\[
\begin{align*}
  f &: (A \land B)A \land B & \quad \text{fst} &: (A \land B)A & \quad \text{inl} &: (A)A \lor B \\
  \text{inl} \circ \text{fst} &: (A \land B)A \lor B \\
  (\text{inl} \circ \text{fst}) \circ f &: (A \land B)A \lor B
\end{align*}
\] (4)

For different permutations of cut we have different yet equivalent proof objects: \((\text{inl} \circ (\text{fst} \circ f)) = ((\text{inl} \circ \text{fst}) \circ f)\).
Conclusion

Outline

1 Introduction and Motivation
2 Preliminary Notes
3 Composition of Deductions
4 Conclusion
contrary to Došen’s claims, the propositions-as-types paradigm does not favour categorical proofs over inferences

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

from these hypothetical judgments we can derive higher-order judgments that we can compose and keep track of their associativity
Conclusion

Thank you for your attention.
Conclusion

References I

Combinatory Logic.

\{Automath\}, a language for mathematics.
Technical report, Department of Mathematics, Eindhoven University of Technology.

Deductive Completeness.

Abstraction and application in adjunction.

Inferential Semantics.

On the Paths of Categories.
Conclusion

References II

The formulae-as-types notion of construction.

A Survey of Proof Theory II.

Functional completeness of cartesian categories.

Introduction to higher order categorical logic.
Cambridge University Press.

Bibliopolis, Napoli.

Martin-Löf’s type theory, Handbook of logic in computer science: Volume 5: Logic and algebraic methods.
Oxford University Press, Oxford.

Composition of Deductions within the Propositions as Types Paradigm.
Conclusion

References III

The Philosophical Position of Proof Theory.