Jumbo λ-Calculus

Paul Blain Levy

University of Birmingham

Abstract. We make an argument that, for any study involving computational effects such as divergence or continuations, the traditional syntax of simply typed lambda-calculus cannot be regarded as canonical, because standard arguments for canonicity rely on isomorphisms that may not exist in an effectful setting. To remedy this, we define a "jumbo lambda-calculus" that fuses the traditional connectives together into more general ones, so-called "jumbo connectives". We provide two pieces of evidence for our thesis that the jumbo formulation is advantageous.
Firstly, we show that the jumbo lambda-calculus provides a "complete" range of connectives, in the sense of including every possible connective that, within the beta-eta theory, possesses a reversible rule.
Secondly, in the presence of effects, we show that there is no decomposition of jumbo connectives into non-jumbo ones that is valid in both call-by-value and call-by-name.

1 Canonicity and Connectives

According to many authors [GLT88,LS86,Pit00], the "canonical" simply typed λ-calculus possesses the following types:

$$A ::= \quad 0 \mid A + A \mid 1 \mid A \times A \mid A \to A \tag{1}$$

There are two variants of this calculus. In some texts [GLT88,LS86] the × connective (type constructor) is a *projection product*, with elimination rules

$$\frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \pi M : A} \qquad \frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \pi' M : B}$$

In other texts [Pit00], × is a *pattern-match product*, with elimination rule

$$\frac{\Gamma \vdash M : A \times B \quad \Gamma, \mathtt{x} : A, \mathtt{y} : B \vdash N : C}{\Gamma \vdash \mathtt{pm}\ M\ \mathtt{as}\ \langle \mathtt{x}, \mathtt{y} \rangle.\ N : C}$$

This choice of five connectives $0, +, 1, \times, \to$ raises some questions.

1. Why not include a *ternary* sum type $+(A, B, C)$?
2. Why not include a type $(A, B) \to C$ of functions that take *two* arguments?
3. Why not include *both* a pattern-match product $A \times B$ *and* a projection product $A \,\Pi\, B$?

In the purely functional setting, these can be answered using Ockham's razor:

1. unnecessary—it would be isomorphic to $(A + B) + C$
2. unnecessary—it would be isomorphic to $(A \times B) \to C$, and to $A \to (B \to C)$
3. unnecessary—they would be isomorphic, so either one suffices.

But these answers are not valid in the presence of effectful constructs, such as recursion or control operators. For example, in a call-by-name language with recursion, $+(A, B, C) \ncong (A + B) + C$ (a point made in [McC96b]), and $A \times B \ncong A \mathbin{\Pi} B$. To see this, consider standard semantics that interprets each type by a pointed cpo. Then $+$ denotes lifted disjoint union, $A \mathbin{\Pi} B$ denotes cartesian product, and $A \times B$ denotes lifted product.

This suggests that, to obtain a canonical formulation of simply typed λ-calculus (suitable for subsequent extension with effects), we should—at least *a priori*—replace Ockham's minimalist philosophy with a maximalist one, treating many combinations of the above connectives as primitive. These combinations are called *jumbo connectives*. But how many connectives must we include to obtain a "complete" range?

A first suggestion might be to include *every* possible combination of the original five as primitive, e.g. a ternary connective γ mapping A, B, C to $(A \to B) \to C$. But this seems unwieldy. We need some criterion of reasonableness that excludes γ but includes all the connectives mentioned above.

We obtain this by noting that each of the above connectives possesses, within the $\beta\eta$ equational theory, a *reversible rule*. For example:

$$\frac{\Gamma, A \vdash B}{\overline{\Gamma \vdash A \to B}} \qquad\qquad \frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\overline{\Gamma, A + B \vdash C}}$$

The rule for $A \to B$ means that we can turn each inhabitant of $\Gamma, A \vdash B$ into an inhabitant of $\Gamma \vdash A \to B$, and vice versa, and these two operations are inverse (up to $\beta\eta$-equality). The rule for $A + B$ is understood similarly. Note also that, in these rules, every part of the conclusion other than the type being introduced appears in each premise. Informally, we shall say that a connective is "$\{0, +, 1, \times, \to\}$-like", when, in the presence of $\beta\eta$, it possesses such a reversible rule. In this paper, we introduce a calculus called "jumbo λ-calculus", and show that it contains every $\{0, +, 1, \times, \to\}$-like connective.

As stated above, our main argument for the necessity of jumbo connectives in the effectful setting is that suggested decompositions are not *a priori* valid, but in Sect. 4 we take this further by showing that, *a posteriori*, they do not have a decomposition that is valid in both CBV and CBN.

Related work Both our arguments for jumbo connectives (invalidity of decompositions, possession of a reversible rule) have arisen in ludics [Gir01].

1.1 Infinitary Variant

Frequently, in semantics, one wishes to study infinitary calculi with countable sum types and countable product types. (The latter are necessarily projection products.) We therefore say that a connective is "$\{0, +, \sum_{i \in \mathbb{N}}, 1, \times, \prod_{i \in \mathbb{N}}, \to\}$-like" when it possesses a reversible rule with countably many premises. By contrast, a $\{0, +, 1, \times, \to\}$-like connective is required to have a reversible rule with finitely many premises.

We shall define an *infinitary* jumbo λ-calculus, as well as the finitary one, and show that the former contains every $\{0, +, \sum_{i \in \mathbb{N}}, 1, \times, \prod_{i \in \mathbb{N}}, \to\}$-like connective.

2 Jumbo λ-calculus

Jumbo λ-calculus is a calculus of *tuples* and *functions*.

2.1 Tuples

A tuple in jumbo λ-calculus has several components; the first component is a tag and the rest are terms. (We often write tags with a # symbol to avoid confusion with identifiers.) An example of a tuple type is

$$\begin{array}{l} \boxed{\sum} \{ \\ \quad \#\mathsf{a}.\ \mathtt{int}, \mathtt{bool} \\ \quad \#\mathsf{b}.\ \mathtt{bool}, \mathtt{int}, \mathtt{bool} \\ \quad \#\mathsf{c}.\ \mathtt{int} \\ \} \end{array} \tag{2}$$

This contains tuples such as $\langle \#\mathsf{a}, 17, \mathtt{false} \rangle$ and $\langle \#\mathsf{b}, \mathtt{true}, 5, \mathtt{true} \rangle$. The type (3) can *roughly* be thought of as an indexed sum of finite products:

$$\begin{array}{l} \sum \{ \\ \quad \#\mathsf{a}.\ (\mathtt{int} \times \mathtt{bool}) \\ \quad \#\mathsf{b}.\ (\mathtt{bool} \times \mathtt{int} \times \mathtt{bool}) \\ \quad \#\mathsf{c}.\ \mathtt{int} \\ \} \end{array} \tag{3}$$

But whether (2) and (3) are actually isomorphic is a matter for investigation below—not something we may assume *a priori*.

If M is a term of the above type, we can pattern-match it:

$$\begin{array}{ll} \multicolumn{2}{l}{\mathtt{pm}\ M\ \mathtt{as}\ \{} \\ \quad \langle \#\mathsf{a}, \mathtt{x}, \mathtt{y} \rangle. & N \\ \quad \langle \#\mathsf{b}, \mathtt{x}, \mathtt{y}, \mathtt{z} \rangle. & P \\ \quad \langle \#\mathsf{c}, \mathtt{w} \rangle. & Q \\ \} & \end{array}$$

where N,P and Q all have the same type.

2.2 Functions

A function in jumbo λ-calculus is applied to several arguments; the first argument is a tag, and the rest are terms. An example of a function type is

$$\boxed{\prod}\{\\ \quad \#\mathsf{a}.\ \mathtt{int}, \mathtt{int}, \mathtt{int} \vdash \mathtt{bool} \\ \quad \#\mathsf{b}.\ \mathtt{int}, \mathtt{bool} \vdash \mathtt{int} \\ \quad \#\mathsf{c}.\ \mathtt{bool}, \mathtt{int} \vdash \mathtt{int} \\ \} \tag{4}$$

An example function of this type is

$$\lambda\{\\ \quad (\#\mathsf{a}, \mathtt{x}, \mathtt{y}, \mathtt{z}).\ \mathtt{x} > (\mathtt{y} + \mathtt{z}) \\ \quad (\#\mathsf{b}, \mathtt{x}, \mathtt{y}).\quad \mathtt{if}\ \mathtt{y}\ \mathtt{then}\ \mathtt{x} + 5\ \mathtt{else}\ \mathtt{x} + 7 \\ \quad (\#\mathsf{c}, \mathtt{x}, \mathtt{y}).\quad \mathtt{y} + 1 \\ \} \tag{5}$$

Applying this to arguments $(\#\mathsf{a}, M, N, P)$ gives a boolean, whereas applying it to arguments $(\#\mathsf{b}, N, N')$ gives an integer. (Note the use of () for multiple arguments, and $\langle\rangle$ for tuple formation.) The type (4) can roughly be thought of as an indexed product of function types:

$$\prod\{\\ \quad \#\mathsf{a}.\ (\mathtt{int} \to (\mathtt{int} \to (\mathtt{int} \to \mathtt{bool}))) \\ \quad \#\mathsf{b}.\ (\mathtt{int} \to (\mathtt{bool} \to \mathtt{int})) \\ \quad \#\mathsf{c}.\ (\mathtt{bool} \to (\mathtt{int} \to \mathtt{int})) \\ \} \tag{6}$$

But again, we cannot assume *a priori* that (4) and (6) are isomorphic.

2.3 Summary

The types and terms of jumbo λ-calculus are shown in Fig. 1. Here, I ranges over all finite sets (for the finitary variant) or over all countable sets (for the infinitary variant), $\overrightarrow{A}$ indicates a finite sequence of types, $|\overrightarrow{A}|$ is its length, and $\$n$ (for $n \in \mathbb{N}$) is the set $\{0, \ldots, n-1\}$. As in, e.g., [Win93], we include a construct `let` to make a binding, although this can be desugared in various ways.

Types $A ::= \boxed{\sum}\ \{\overrightarrow{A}_i\}_{i \in I} \ \mid\ \boxed{\prod}\ \{\overrightarrow{A}_i \vdash A_i\}_{i \in I}$

Terms

$$\frac{}{\Gamma, \mathtt{x} : A, \Gamma' \vdash \mathtt{x} : A} \qquad \frac{\Gamma \vdash N : A \quad \Gamma, \mathtt{x} : A \vdash M : B}{\Gamma \vdash \mathtt{let}\ N\ \mathtt{be}\ \mathtt{x}.\ M : B}$$

$$\frac{\hat{\imath} \in I \quad \Gamma \vdash N_j : A_{\hat{\imath}j}\ \ (\forall j \in \$|\overrightarrow{A}_{\hat{\imath}}|)}{\Gamma \vdash \langle \hat{\imath}, \overrightarrow{N} \rangle : \boxed{\sum}\ \{\overrightarrow{A}_i\}_{i \in I}} \qquad \frac{\Gamma \vdash N : \boxed{\sum}\ \{\overrightarrow{A}_i\}_{i \in I} \quad \Gamma, \overrightarrow{\mathtt{x}} : \overrightarrow{A}_i \vdash M_i : B\ \ (\forall i \in I)}{\Gamma \vdash \mathtt{pm}\ N\ \mathtt{as}\ \{\langle i, \overrightarrow{\mathtt{x}} \rangle . M_i\}_{i \in I} : B}$$

$$\frac{\Gamma, \overrightarrow{\mathtt{x}} : \overrightarrow{A}_i \vdash M_i : B_i\ \ (\forall i \in I)}{\Gamma \vdash \lambda \{(i, \overrightarrow{\mathtt{x}}).M_i\}_{i \in I} : \boxed{\prod}\ \{\overrightarrow{A}_i \vdash B_i\}_{i \in I}} \qquad \frac{\Gamma \vdash M : \boxed{\prod}\ \{\overrightarrow{A}_i \vdash B_i\}_{i \in I} \quad \hat{\imath} \in I \quad \Gamma \vdash N_j : A_{\hat{\imath}j}\ \ (\forall j \in \$|\overrightarrow{A}_{\hat{\imath}}|)}{\Gamma \vdash M(\hat{\imath}, \overrightarrow{N}) : B_{\hat{\imath}}}$$

Fig. 1. Syntax Of Jumbo λ-calculus

2.4 Jumbo-arities

Many traditional connectives are special cases of the jumbo connectives:

type	comments	expressed as
$A + B$		$\boxed{\sum}\{\#\mathsf{left}.A, \#\mathsf{right}.B\}$
$\sum_{i \in I} A_i$		$\boxed{\sum}\{A_i\}_{i \in I}$
$A \times B$	pattern-match product	$\boxed{\sum}\{\#\mathsf{sole}.A, B\}$
$\times(\overrightarrow{A})$	n-ary pattern-match product	$\boxed{\sum}\{\#\mathsf{sole}.\overrightarrow{A}\}$
$A\ \Pi\ B$	projection product	$\boxed{\prod}\{\#\mathsf{left}. \vdash A, \#\mathsf{right}. \vdash B\}$
$\prod_{i \in I} A_i$	I-ary projection product	$\boxed{\prod}\{\vdash A_i\}_{i \in I}$
$A \to B$	type of functions with one argument	$\boxed{\prod}\{\#\mathsf{sole}.A \vdash B\}$
$(\overrightarrow{A}) \to B$	type of functions with n arguments	$\boxed{\prod}\{\#\mathsf{sole}.\overrightarrow{A} \vdash B\}$
`bool`		$\boxed{\sum}\{\#\mathsf{true}.\epsilon, \#\mathsf{false}.\epsilon\}$
ground_I	ground type with I elements	$\boxed{\sum}\{\epsilon\}_{i \in I}$
TA	studied in call-by-value setting [Mog89]	$\boxed{\prod}\{\#\mathsf{sole}. \vdash A\}$
LA	studied in call-by-name setting [McC96a]	$\boxed{\sum}\{\#\mathsf{sole}.A\}$

To make this more systematic, define a *jumbo-arity* to be a countable family of natural numbers $\{n_i\}_{i \in I}$. Then both $\boxed{\sum}$ and $\boxed{\prod}$ provide a family of connectives, indexed by jumbo-arities, as follows.

- Each jumbo-arity $\{n_i\}_{i \in I}$, determines a connective $\boxed{\sum}_{\{n_i\}_{i \in I}}$ of arity $\sum_{i \in I} n_i$. Given types $\{A_{ij}\}_{i \in I, j \in \$ n_i}$, it constructs the type $\boxed{\sum}\ \{A_{i0}, \ldots, A_{i(n_i - 1)}\}_{i \in I}$.
- Each jumbo-arity $\{n_i\}_{i \in I}$, determines a connective $\boxed{\prod}_{\{n_i\}_{i \in I}}$ of arity $\sum_{i \in I}(n_i + 1)$. Given types $\{A_{ij}\}_{i \in I, j \in \$ n_i}$ and types $\{B_i\}_{i \in I}$, it constructs the type $\boxed{\prod}\ \{A_{i0}, \ldots, A_{i(n_i - 1)} \vdash B_i\}_{i \in I}$.

Corresponding to the above instances, we have

connective	arity	expressed as
$+$	2	$\boxed{\sum}_{\{\#\mathsf{left}.1,\#\mathsf{right}.1\}}$
$\sum_{i\in I}$	I	$\boxed{\sum}_{\{1\}_{i\in I}}$
$\times$	2	$\boxed{\sum}_{\{\#\mathsf{sole}.2\}}$
$\times$	n	$\boxed{\sum}_{\{\#\mathsf{sole}.n\}}$
$\sqcap$	2	$\boxed{\prod}_{\{\#\mathsf{left}.0,\#\mathsf{right}.0\}}$
$\prod_{i\in I}$	I	$\boxed{\prod}_{\{0\}_{i\in I}}$
$\to$	2	$\boxed{\prod}_{\{\#\mathsf{sole}.1\}}$
$\to$	$n+1$	$\boxed{\prod}_{\{\#\mathsf{sole}.n\}}$
`bool`	0	$\boxed{\sum}_{\{\#\mathsf{true}.0,\#\mathsf{false}.0\}}$
ground_I	0	$\boxed{\sum}_{\{0\}_{i\in I}}$
T	1	$\boxed{\prod}_{\{\#\mathsf{sole}.0\}}$
L	1	$\boxed{\sum}_{\{\#\mathsf{sole}.1\}}$

3 The $\beta\eta$-theory of Jumbo λ-calculus

3.1 Laws and Isomorphisms

In the absence of computational effects, the most natural equational theory for the jumbo λ-calculus is the $\beta\eta$-theory, displayed in Fig. 2.

A *$\beta\eta$-isomorphism* $A \xrightarrow{\cong} B$ is a pair of terms $\mathtt{y} : A \vdash \alpha : B$ and $\mathtt{z} : B \vdash \alpha^{-1} : A$ such that $\alpha^{-1}[\alpha/\mathtt{z}] = \mathtt{y}$ and $\alpha[\alpha^{-1}/\mathtt{y}] = \mathtt{z}$ is provable up to $\beta\eta$-equality. We identify α and α' when $\alpha = \alpha'$ is provable.

The $\beta\eta$-theory gives non-jumbo decompositions and other isomorphisms, e.g.

$$\boxed{\sum}\,\{A_{i0},\ldots,A_{i(n_i-1)}\}_{i\in I} \cong \textstyle\sum_{i\in I}(A_{i0}\times\cdots\times A_{i(n_i-1)})$$
$$\boxed{\prod}\,\{A_{i0},\ldots,A_{i(n_i-1)} \vdash B_i\}_{i\in I} \cong \textstyle\prod_{i\in I}(A_{i0}\to\cdots A_{i(n_i-1)}\to B_i)$$
$$\times(\overrightarrow{A}) \cong \sqcap(\overrightarrow{A})$$
$$TA \cong A \cong LA$$

So the $\beta\eta$-theory makes the jumbo λ-calculus equivalent to that of Sect. 1.

3.2 Reversible Rules

Our next task is to make precise the notion of reversible rule from Sect. 1.

Definition 1 1. For a sequent $s = \Gamma \vdash A$ (i.e. a pair of a context Γ and a type A), we write $\mathsf{inhab}\, s$ for the set of terms (modulo $\beta\eta$-equality) inhabiting s.
2. For a countable family of sequents $S = \{s_i\}_{i\in I}$, we write $\mathsf{inhab}\, S$ for $\prod_{i\in I} s_i$.

β-laws

$$\frac{\Gamma \vdash N : A \quad \Gamma, \mathtt{x} : A \vdash M : B}{\Gamma \vdash \mathtt{let}\ N\ \mathtt{be}\ \mathtt{x}.\ M = M[N/\mathtt{x}] : B}$$

$$\frac{\hat{\imath} \in I \quad \Gamma \vdash N_j : A_{\hat{\imath}j} \ \ (\forall j \in \$|\overrightarrow{A}_{\hat{\imath}}|) \quad \Gamma, \overrightarrow{\mathtt{x}} : \overrightarrow{A}_i \vdash M_i : B \ \ (\forall i \in I)}{\Gamma \vdash \mathtt{pm}\ \langle \hat{\imath}, \overrightarrow{N} \rangle\ \mathtt{as}\ \{\langle i, \overrightarrow{\mathtt{x}} \rangle.M_i\}_{i \in I} = M_{\hat{\imath}}[\overrightarrow{N/\mathtt{x}}] : B_{\hat{\imath}}}$$

$$\frac{\Gamma, \overrightarrow{\mathtt{x}} : \overrightarrow{A}_i \vdash M : B_i \ \ (\forall i \in I) \quad \hat{\imath} \in I \quad \Gamma \vdash N_j : A_{\hat{\imath}j} \ \ (\forall j \in \$|\overrightarrow{A}_{\hat{\imath}}|)}{\Gamma \vdash \lambda\{(i, \overrightarrow{\mathtt{x}}).M_i\}_{i \in I}(\hat{\imath}, \overrightarrow{N}) = M_{\hat{\imath}}[\overrightarrow{N/\mathtt{x}}] : B_{\hat{\imath}}}$$

η-laws

$$\frac{\Gamma \vdash N : \boxed{\sum} \{\overrightarrow{A}_i\}_{i \in I} \quad \Gamma, \mathtt{z} : \boxed{\sum} \{\overrightarrow{A}_i\}_{i \in I} \vdash M : B}{\Gamma \vdash M[N/\mathtt{z}] = \mathtt{pm}\ N\ \mathtt{as}\ \{\langle i, \overrightarrow{\mathtt{x}} \rangle.M[\langle i, \overrightarrow{\mathtt{x}} \rangle/\mathtt{z}]\}_{i \in I} : B} \ \overrightarrow{\mathtt{x}} \text{ fresh for } \Gamma$$

$$\frac{\Gamma \vdash M : \boxed{\prod} \{\overrightarrow{A}_i \vdash B_i\}_{i \in I}}{\Gamma \vdash M = \lambda\{(i, \overrightarrow{\mathtt{x}}).M(i, \overrightarrow{\mathtt{x}})\}_{i \in I} : \boxed{\prod} \{\overrightarrow{A}_i \vdash B_i\}_{i \in I}} \ \overrightarrow{\mathtt{x}} \text{ fresh for } \Gamma$$

Fig. 2. The $\beta\eta$ Equational Theory For Jumbo λ-calculus

3. A *rule* from sequent family S to sequent family S' is a function from $\mathsf{inhab}\, S$ to $\mathsf{inhab}\, S'$.

□

The reversible rules for $\to$ and $+$ shown in Sect. 1 are given for all Γ, and, in the case of $+$, for all C. Furthermore, they are "natural", as we now explain.

Definition 2 1. [Lawvere] A *substitution* from a context $\Gamma = A_0, \ldots, A_{m-1}$ to a context Γ' is a sequence of terms $M_0, \ldots, M_{m-1}$ where $\Gamma' \vdash M_i : A_i$ for each $i \in \$m$. As usual, such a morphism induces a substitution function q^* from terms $\Gamma, \Delta \vdash B$ to terms $\Gamma', \Delta \vdash B$.
2. Any term $\Gamma, \mathtt{y} : C \vdash P : C'$ gives rise to a function $P^\dagger$ from terms inhabiting $\Gamma, \Delta \vdash C$ to terms inhabiting $\Gamma, \Delta \vdash C'$, where $P^\dagger N = P[N/\mathtt{y}]$.

□

The $\to$ and $+$ reversible rules are *natural in* Γ in the sense that they commute with q^*, up to $\beta\eta$-equality, for any context morphism $\Gamma' \xrightarrow{q} \Gamma$. (Actually, they commute up to syntactic equality, but that is not significant here.) The $+$ reversible rule is also *natural in* C in the sense that it commutes with $P^\dagger$, up to $\beta\eta$-equality, for any term $\Gamma, \mathtt{y} : C \vdash P : C'$.

Definition 3 A *reversible rule* for a type B, in an equational theory, is a rule r with a single conclusion, such that

- r is a bijection

- the conclusion contains a single occurrence of B (adjacent to $\vdash$, let us say)
- the rest of the conclusion is arbitrary, appears in every premise, and the rule is natural in it.

In detail, either

reversible left rule the conclusion is $\Gamma, B \vdash C$, every premise contains $\Gamma \vdash C$—i.e. is of the form $\Gamma, \Delta \vdash C$—and r is natural in Γ and C, or

reversible right rule the conclusion is $\Gamma \vdash B$, every premise contains $\Gamma \vdash$—i.e. is of the form $\Gamma, \Delta \vdash B'$—and r is natural in Γ.

$\square$

Definition 4 We associate to the type $\boxed{\sum}\{\overrightarrow{A}_i\}_{i\in I}$ the reversible left rule

$$\frac{\Gamma, \overrightarrow{\mathtt{x}} : \overrightarrow{A}_i \vdash C \quad (\forall i \in I)}{\Gamma, \mathtt{y} : \boxed{\sum}\{\overrightarrow{A}_i\}_{i\in I} \vdash C} \qquad \begin{array}{lcl} \{M_i\}_{i\in I} & \mapsto & \mathtt{pm}\ \mathtt{y}\ \mathtt{as}\ \{\langle i, \overrightarrow{\mathtt{x}}\rangle.M_i\}_{i\in I} \\ N & \mapsto & \{N[\langle i, \overrightarrow{\mathtt{x}}\rangle/\mathtt{y}]\}_{i\in I} \end{array}$$

We associate to the type $\boxed{\prod}\{\overrightarrow{A}_i \vdash B_i\}_{i\in I}$ the reversible right rule

$$\frac{\Gamma, \overrightarrow{\mathtt{x}} : \overrightarrow{A}_i \vdash B_i \quad (\forall i \in I)}{\Gamma \vdash \boxed{\prod}\{\overrightarrow{A}_i \vdash B_i\}_{i\in I}} \qquad \begin{array}{lcl} \{M_i\}_{i\in I} & \mapsto & \lambda\{(i, \overrightarrow{\mathtt{x}}).M_i\}_{i\in I} \\ N & \mapsto & N(i, \overrightarrow{\mathtt{x}}) \end{array}$$

$\square$

Definition 5 Given a reversible rule r for A, and an $\beta\eta$-isomorphism $A \xrightarrow{\cong} B$ comprised of $\mathtt{y} : A \vdash \alpha : B$ and $\mathtt{z} : B \vdash \alpha^{-1} : A$, we define a reversible rule r_α for B.

- If r is left, with conclusion $\Gamma, \mathtt{y} : A \vdash C$, then r_α has conclusion $\Gamma, \mathtt{z} : B \vdash C$. It maps a to $r(a)[\alpha^{-1}/\mathtt{y}]$, and its inverse maps N to $r^{-1}(N[\alpha/\mathtt{z}])$.
- If r is right, with conclusion $\Gamma \vdash A$, then r_α has conclusion $\Gamma \vdash B$. It maps a to $\alpha[r(a)/\mathtt{y}]$ and its inverse maps N to $r^{-1}(\alpha^{-1}[N/\mathtt{z}])$.

$\square$

We can now state the main technical property of jumbo λ-calculus:

Proposition 1 Let s be a reversible rule in the $\beta\eta$-theory of jumbo λ-calculus. Then s is r_α, where r is one of the rules in Def. 4 and α a $\beta\eta$-isomorphism; and r and α are unique. $\square$

Proof Suppose s is left, with conclusion $\Gamma, \mathtt{z} : B \vdash C$. Call the set indexing its premises I. For each $i \in I$, the ith premise must be of the form $\Gamma, \overrightarrow{\mathtt{x}} : \overrightarrow{A}_i \vdash C$. Set A to be the type $\boxed{\sum}\{\overrightarrow{A}_i\}_{i\in I}$, and r to be the reversible rule that Def. 4 associates to this type. That is clearly is the only possibility for r.

The rest is a syntactic version of the (indexed) Yoneda lemma. Define

– $\mathtt{y} : A \vdash \alpha : B$ to be $rs^{-1}(\mathtt{z} : B \vdash \mathtt{z} : B)$
– $\mathtt{z} : B \vdash \alpha^{-1} : A$ to be $sr^{-1}(\mathtt{y} : A \vdash \mathtt{y} : A)$.

We claim that

$$sr^{-1}(\Gamma, \mathtt{y} : A \vdash M : C) = M[\alpha^{-1}/\mathtt{y}] \qquad (7)$$
$$rs^{-1}(\Gamma, \mathtt{z} : B \vdash N : C) = N[\alpha/\mathtt{z}] \qquad (8)$$

For (7), we note that $M = M^{\dagger} k_{\Gamma}^{*}(\mathtt{y} : A \vdash \mathtt{y} : A)$. (Here k_{Γ} means the unique substitution from the empty context to Γ.) Hence the LHS is $sr^{-1}(M^{\dagger} k_{\Gamma}^{*}(\mathtt{y}))$. By naturality of s and r, this is $M^{\dagger} k_{\Gamma}^{*}(sr^{-1}(\mathtt{y}))$, which is $M^{\dagger} k_{\Gamma}^{*}(\alpha^{-1})$, the RHS. (8) is similar. Setting M to be α in (7) gives $\mathtt{z} = \alpha[\alpha^{-1}/\mathtt{y}]$, and similarly $\mathtt{y} = \alpha^{-1}[\alpha/\mathtt{z}]$. Setting M to be $r(a)$ in (7) gives $s = r_{\alpha}$. For uniqueness, $s = r_{\beta}$ implies

$$\alpha = rr_{\beta}^{-1}(\mathtt{z} : B \vdash \mathtt{z} : B) = rr^{-1}(\mathtt{z}[\beta/\mathtt{z}]) = \beta$$

The argument in the case that s is right is similar but easier.

$\square$

Thus $\boxed{\sum}$ and $\boxed{\prod}$ are the most general $\{0, +, \sum_{i \in I}, 1, \times, \prod_{i \in I}, \rightarrow\}$-like connectives, and the infinitary jumbo λ-calculus is greatest among calculi consisting of such connectives. Similarly, $\boxed{\sum}$ and $\boxed{\prod}$ with finite tag set are the most general $\{0, +, 1, \times, \rightarrow\}$-like connectives, and the finitary jumbo λ-calculus is greatest among calculi consisting of such connectives.

4 λ-Calculus Plus Computational Effects

4.1 Operational Semantics

In Sect. 4.1–4.2, we adapt standard material from e.g. [Win93] to the setting of jumbo λ-calculus. As a very simple example of a computational effect, let us consider divergence. So we add to the jumbo λ-calculus the typing rule

$$\frac{}{\Gamma \vdash \mathtt{diverge} : B}$$

where B may be any type. The $\beta\eta$-theory is inconsistent in the presence of a closed term of type 0, so we discard it. Our statement that each connective is $\{0, +, \sum_{i \in \mathbb{N}}, 1, \times, \prod_{i \in \mathbb{N}}, \rightarrow\}$-like means that *in the presence of* $\beta\eta$ it has a reversible rule. Since we have now discarded $\beta\eta$, these rules are lost.

We consider two languages with this syntax: call-by-name and call-by-value. As usual, each is defined by an operational semantics that maps closed terms to a special class of closed terms called *terminal terms*. We define this by an interpreter in Fig. 3. The metalanguage for the interpreter (written in italics) is first-order and recursive, containing the following constructs:

rec f lambda	for a recursive definition of a function f
P to D. Q	to mean: first evaluate P, then, if that gives D, evaluate Q
$\overrightarrow{P\ to\ D}$*. Q*	to abbreviate P_0 *to* $D_0 \ldots . P_{n-1}$ *to* D_{n-1}. Q.

$$\textbf{Terminal Terms} \begin{cases} \textbf{CBN} & \text{Closed terms of the form } \langle \hat{\imath}, \overrightarrow{M} \rangle \text{ or } \lambda\{(i, \overrightarrow{\mathtt{x}}).M_i\}_{i \in I} \\ \textbf{CBV} & \text{Inductively defined by } T ::= \ \langle \hat{\imath}, \overrightarrow{T} \rangle \mid \lambda\{(i, \overrightarrow{\mathtt{x}}).M_i\}_{i \in I} \end{cases}$$

CBN interpreter *rec cbn lambda*{

$\mathtt{let}\ N\ \mathtt{be}\ \mathtt{x}.\ M$	.	$\mathit{cbn}\ M[N/\mathtt{x}]$
$\langle \hat{\imath}, \overrightarrow{N} \rangle$	.	$\mathit{return}\ \langle \hat{\imath}, \overrightarrow{N} \rangle$
$\mathtt{pm}\ N\ \mathtt{as}\ \{\langle i, \overrightarrow{\mathtt{x}} \rangle.M_i\}_{i \in I}$	.	$(\mathit{cbn}\ N)\ \mathit{to}\ \langle \hat{\imath}, \overrightarrow{N} \rangle.\ \mathit{cbn}\ M_{\hat{\imath}}[\overrightarrow{N/\mathtt{x}}]$
$\lambda\{(i, \overrightarrow{\mathtt{x}}).M_i\}_{i \in I}$	.	$\mathit{return}\ \lambda\{(i, \overrightarrow{\mathtt{x}}).M_i\}_{i \in I}$
$M(\hat{\imath}, \overrightarrow{N})$	.	$(\mathit{cbn}\ M)\ \mathit{to}\ \lambda\{(i, \overrightarrow{\mathtt{x}}).M_i\}_{i \in I}.\ \mathit{cbn}\ M_{\hat{\imath}}[\overrightarrow{N/\mathtt{x}}]$
$\mathtt{diverge}$	.	*diverge*

}

CBV (left-to-right) interpeter *rec cbv lambda*{

$\mathtt{let}\ N\ \mathtt{be}\ \mathtt{x}.\ M$	.	$(\mathit{cbv}\ N)\ \mathit{to}\ T.\ \mathit{cbv}\ M[T/\mathtt{x}]$
$\langle \hat{\imath}, \overrightarrow{N} \rangle$	.	$\overrightarrow{(\mathit{cbv}\ N)\ \mathit{to}\ T}.\ \mathit{return}\ \langle \hat{\imath}, \overrightarrow{T} \rangle$
$\mathtt{pm}\ N\ \mathtt{as}\ \{\langle i, \overrightarrow{\mathtt{x}} \rangle.M_i\}_{i \in I}$	.	$(\mathit{cbv}\ N)\ \mathit{to}\ \langle \hat{\imath}, \overrightarrow{T} \rangle.\ \mathit{cbv}\ M_{\hat{\imath}}[\overrightarrow{T/\mathtt{x}}]$
$\lambda\{(i, \overrightarrow{\mathtt{x}}).M_i\}_{i \in I}$	.	$\mathit{return}\ \lambda\{(i, \overrightarrow{\mathtt{x}}).M_i\}_{i \in I}$
$M(\hat{\imath}, \overrightarrow{N})$	.	$(\mathit{cbv}\ M)\ \mathit{to}\ \lambda\{(i, \overrightarrow{\mathtt{x}}).M_i\}_{i \in I}.\ \overrightarrow{(\mathit{cbv}\ N)\ \mathit{to}\ T}.\ \mathit{cbv}\ M_{\hat{\imath}}[\overrightarrow{T/\mathtt{x}}]$
$\mathtt{diverge}$	.	*diverge*

}

Fig. 3. CBN and (left-to-right) CBV interpreters

Remark 1. Notice the consequences of the call-by-value semantics for the two binary products. A terminal term in $A \times B$ (the pattern-match product) is $\langle T, T' \rangle$, where T and T' are terminal. But, because we do not evaluate under λ, a terminal term in $A \sqcap B$ (the projection product) is $\lambda\{0.M, 1.N\}$, where M and N need not be terminal. This differs from the formulation in [Win93]. □

We write $M \Downarrow_{\mathbf{CBN}} T$ to mean that M evaluates to T in CBN, which can be defined inductively in the usual way. Otherwise M diverges and we write $M \Uparrow_{\mathbf{CBN}}$. Similarly for CBV.

For call-by-value, we inductively define *values*: $V ::= \ \mathtt{x} \mid \langle i, \overrightarrow{V} \rangle \mid \lambda\{(i, \overrightarrow{\mathtt{x}}).M_i\}_{i \in I}$

4.2 Denotational Semantics

We extend the cpo semantics for CBN and CBV in [Win93] as follows.

In the call-by-name language, a type denotes a cpo with least element:

$$[\![\boxed{\textstyle\sum} \{A_{i\,0}, \ldots, A_{i\,(n_i-1)}\}_{i \in I}]\!] = \Big(\sum_{i \in I}([\![A_{i\,0}]\!] \times \cdots \times [\![A_{i\,(n_i-1)}]\!])\Big)_{\bot}$$

$$[\![\boxed{\textstyle\prod} \{A_{i\,0}, \ldots, A_{i\,n_i-1} \vdash B_i\}_{i \in I}]\!] = \prod_{i \in I}([\![A_{i\,0}]\!] \to \cdots \to [\![A_{i\,(n_i-1)}]\!] \to [\![B_i]\!])$$

A context $\Gamma = A_0, \ldots, A_{n-1}$ denotes the cpo $[\![A_0]\!] \times \cdots \times [\![A_{n-1}]\!]$, and a term $\Gamma \vdash M : B$ denotes a continuous function $[\![\Gamma]\!] \xrightarrow{[\![M]\!]} [\![B]\!]$.

In the call-by-value language, a type denotes a cpo:

$$[\![\boxed{\sum}\{A_{i\,0},\ldots,A_{i\,(n_i-1)}\}_{i\in I}]\!] = \sum_{i\in I}([\![A_{i\,0}]\!]\times\cdots\times[\![A_{i\,(n_i-1)}]\!])$$

$$[\![\boxed{\prod}\{A_{i\,0},\ldots,A_{i\,(n_i-1)}\vdash B_i\}_{i\in I}]\!] = \prod_{i\in I}([\![A_{i\,0}]\!]\to\cdots\to[\![A_{i\,(n_i-1)}]\!]\to([\![B_i]\!]_\bot))$$

A context $\Gamma = A_0,\ldots,A_{n-1}$ denotes $[\![A_0]\!]\times\cdots\times[\![A_{n-1}]\!]$, and a term $\Gamma\vdash M : B$ denotes a continuous function $[\![\Gamma]\!]\xrightarrow{[\![M]\!]}[\![B]\!]_\bot$. Each value $\Gamma\vdash V : B$ has another denotation $[\![\Gamma]\!]\xrightarrow{[\![V]\!]^{\mathsf{val}}}[\![B]\!]$ such that $[\![V]\!]\rho = \mathsf{up}\,([\![V]\!]^{\mathsf{val}}\rho)$ for all $\rho\in[\![\Gamma]\!]$.

The detailed semantics of CBN terms and of CBV terms and values are obvious and omitted. For both languages, we prove a substitution lemma, then show that $M \Downarrow T$ implies $[\![M]\!] = [\![T]\!]$, and $M \Uparrow$ implies $[\![M]\!] = \bot$, as in [Win93].

4.3 Invalidity Of Decompositions

We say that types A and B are

- *cpo-isomorphic in CBN* when $[\![A]\!]_{\mathbf{CBN}}$ and $[\![B]\!]_{\mathbf{CBN}}$ are isomorphic cpos
- *cpo-isomorphic in CBV* when $[\![A]\!]_{\mathbf{CBV}}$ and $[\![B]\!]_{\mathbf{CBV}}$ are isomorphic cpos.

This is very liberal: e.g., 1_Π and 0 are cpo-isomorphic in CBN, though not isomorphic in other CBN models. But the purpose of this section is to establish *non*-isomorphisms, so that is good enough.

We begin by investigating the most obvious decompositions.

Proposition 2 The following decompositions are cpo-isomorphisms in CBN but not CBV:

$$\Pi(A_0,\ldots,A_{n-1}) \cong A_0\,\Pi\,A_1\cdots\Pi\,A_{n-1}$$
$$\boxed{\sum}\{\overrightarrow{A}_i\}_{i\in I} \cong \textstyle\sum_{i\in I}\Pi\,(\overrightarrow{A}_i)$$
$$(A_0,\ldots,A_{n-1})\to B \cong A_0\to A_1\to\cdots\to A_{n-1}\to B$$
$$(A_0,\ldots,A_{n-1})\to B \cong (A_0\,\Pi\cdots\Pi\,A_{n-1})\to B$$
$$\boxed{\prod}\{\overrightarrow{A}_i\vdash B_i\}_{i\in I} \cong \textstyle\prod_{i\in I}((\overrightarrow{A}_i)\to B_i)$$

The following decompositions are cpo-isomorphisms in CBV but not CBN:

$$+(A_0,\ldots,A_{n-1}) \cong A_0 + A_1\cdots + A_{n-1}$$
$$\times(A_0,\ldots,A_{n-1}) \cong A_0\times A_1\cdots\times A_{n-1}$$
$$\boxed{\sum}\{\overrightarrow{A}_i\}_{i\in I} \cong \textstyle\sum_{i\in I}\times\,(\overrightarrow{A}_i)$$
$$(A_0,\ldots,A_{n-1})\to B \cong (A_0\times\cdots\times A_{n-1})\to B$$
$$\boxed{\prod}\{\overrightarrow{A}_i\vdash B_i\}_{i\in\$n} \cong \times_{i\in\$n}((\overrightarrow{A}_i)\to B_i)$$
$$\boxed{\prod}\{\overrightarrow{A}_i\vdash B_i\}_{i\in I} \cong \boxed{\prod}\{\times(\overrightarrow{A}_i)\vdash B_i\}_{i\in I}$$

Some special cases:

			CBV	CBN
$1_\times$	$\cong$	1_Π	yes	no
$\times\overrightarrow{A}$	$\cong$	$\Pi\overrightarrow{A}$	no	no
ground_I	$\cong$	$\sum_{i\in I}1_\times$	yes	no
ground_I	$\cong$	$\sum_{i\in I}1_\Pi$	yes	yes
TA	$\cong$	A	no	yes
LA	$\cong$	A	yes	no

□

Proof For non-isomorphisms: make all the types `bool`, and count elements. □

A stronger statement of non-decomposability is the following. (We omit its proof, which analyzes finite elements.)

Proposition 3 Call the following types of jumbo λ-calculus *non-jumbo*.

$$A ::= \quad \mathsf{ground}_I \mid \textstyle\sum_{i\in I} A_i \mid \times(\overrightarrow{A}) \mid \prod_{i\in I} A_i \mid (\overrightarrow{A}) \to B$$

1. There is no non-jumbo type A such that $\boxed{\sum}\{\#\mathsf{a}.\mathtt{bool}, \mathtt{bool}; \#\mathsf{b}.\mathtt{bool}\}$ is cpo-isomorphic to A in both CBV and CBN.
2. There is no non-jumbo type A such that $\boxed{\prod}\{\#\mathsf{a}.\mathtt{bool} \vdash \mathtt{bool}; \#\mathsf{b}. \vdash \mathtt{bool}\}$ is cpo-isomorphic to A in both CBV and CBN.
3. There is no non-jumbo type A such that $\boxed{\prod}\{T\mathtt{bool} \vdash \mathsf{ground}_{\$n}\}_{n\in\mathbb{N}}$ is cpo-isomorphic to A in CBV.

□

Thus, neither $\boxed{\sum}$ nor $\boxed{\prod}$ has a universally valid decomposition. And in the infinitary CBV setting, $\boxed{\prod}$ cannot be decomposed at all.

References

[Gir01] J.-Y. Girard. Locus solum: From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11(3):301–506, 2001.

[GLT88] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types.* Cambridge Tracts in Theoretical Computer Science 7. Cambridge University Press, 1988.

[LS86] J. Lambek and P. Scott. *Introduction to Higher Order Categorical Logic.* Cambridge University Press, Cambridge, 1986.

[McC96a] G. McCusker. Full abstraction by translation. Proc., 3rd Workshop in Theory and Formal Methods, Imperial College, London., 1996.

[McC96b] G. McCusker. *Games and Full Abstraction for a Functional Metalanguage with Recursive Types.* PhD thesis, University of London, 1996.

[Mog89] E. Moggi. Computational lambda-calculus and monads. In *LICS'89, Proc. 4th Ann. Symp. on Logic in Comp. Sci.*, pages 14–23. IEEE, 1989.

[Pit00] A. M. Pitts. Categorical logic. In *Handbook of Logic in Computer Science, Vol. 5.* Oxford University Press, 2000.

[Win93] G. Winskel. *Formal Semantics of Programming Languages.* MIT Press, 1993.