A refined interpretation of intuitionistic logic by means of atomic polymorphism

José Espírito Santo and Gilda Ferreira

July 16, 2018

Abstract

We study an alternative embedding of IPC into atomic system F whose translation of proofs is based, not on instantiation overflow, but instead on the admissibility of the elimination rules for disjunction and absurdity (where these connectives are defined according to the Russell-Prawitz translation). As compared to the embedding based on instantiation overflow, the alternative embedding works equally well at the levels of provability and preservation of proof identity, but it produces shorter derivations and shorter simulations of reduction sequences. Lambda-terms are employed in the technical development so that the algorithmic content is made explicit, both for the alternative and the original embeddings. The investigation of preservation of proof-reduction steps by the alternative embedding enables the analysis of generation of "administrative" redexes. These are the key, on the one hand, to understand the difference between the two embeddings; on the other hand, to understand whether the final word on the embedding of IPC into atomic system F has been said.

Keywords: Intuitionistic propositional calculus, predicative polymorphism, Russell-Prawitz translation, permutative conversion, instantiation overflow.

1 Introduction

Since 2006, it is known that the intuitionistic propositional calculus $\mathbf{IPC}$ can be embedded into system $\mathbf{F_{at}}$ – the restriction of Girard's polymorphic system $\mathbf{F}$ to *atomic* universal instantiations [1]. Such embedding of $\mathbf{IPC}$ into $\mathbf{F_{at}}$ relies on the Russell-Prawitz's translation of the connectives bottom and disjunction, $\perp := \forall X.X$ and $A \vee B := \forall X.((A \supset X) \wedge (B \supset X)) \supset X$, and on *instantiation overflow* – the possibility of deriving in $\mathbf{F_{at}}$ the instantiation of the above universal formulas by any (not necessarily atomic) formula[1]. This embedding, which we call the *canonical* embedding, works at the levels of provability and proof reduction, with $\beta\eta$-conversions being preserved by the translation [3, 4], and commutative conversions being mapped to $\beta\eta$-equalities [2].

[1]Initially conjunction was not considered as primitive in $\mathbf{F_{at}}$ but in recent publications [4, 5] it has been the case. See the discussion about the advantages of taking $\wedge$ as primitive in the atomic polymorphic system in Section 5 below.

Instantiation overflow may be seen as a proof transformation. For instance, given in $\mathbf{F_{at}}$ a proof M of $A \vee B$ and an arbitrary formula C, there is in $\mathbf{F_{at}}$ another proof $\underline{\texttt{io}}(M, A, B, C)$ of $((A \supset C) \wedge (B \supset C)) \supset C$ obtained from M. In this paper we challenge instantiation overflow as the basis on which the embedding of $\mathbf{IPC}$ into $\mathbf{F_{at}}$ rests. Specifically, we define an alternative translation of proofs of $\mathbf{IPC}$ into proofs of $\mathbf{F_{at}}$ based on the proof transformations that witness the admissibility in $\mathbf{F_{at}}$ of the elimination rules for $\perp$ and $A \vee B$. We develop the alternative translation of proofs and show it produces an embedding of $\mathbf{IPC}$ into $\mathbf{F_{at}}$ that works at the levels of provability and proof reduction as well as the original, canonical embedding.

In addition, instantiation overflow is an immediate corollary of the referred admissibility. For instance, let us see the case of $A \vee B$. Given in $\mathbf{F_{at}}$ a proof M of $A \vee B$ and proofs P (resp. Q) of C depending on $x : A$ (resp. $y : B$), there is in $\mathbf{F_{at}}$ a proof $\underline{\texttt{case}}(M, x.P, y.Q, C)$ of C obtained from M. Then instantiation overflow is derivable, as seen in the following equation written in λ-notation:

$$\underline{\texttt{io}}(M, A, B, C) = \lambda z^{(A \supset C) \wedge (B \supset C)}.\underline{\texttt{case}}(M, x^A.z1x, y^B.z2y, C) \ . \tag{1}$$

Although the alternative embedding enjoys similar properties of preservation of reduction steps, it produces a much more economical simulation, in the sense that fewer reduction steps in the target calculus are needed to simulate each source reduction step. This leads us to analyze the reason for this parsimony, and we conclude that there are redexes in the translation of a proof that correspond to no redex in the original proof, but instead are created by the translation itself.

An example of such an "administrative" redex is seen in the following equation:

$$\underline{\texttt{case}}(M, x^A.P, y^B.Q, C) = \underline{\texttt{io}}(M, A, B, C)\langle \lambda x^A.P, \lambda y^B.Q \rangle \ . \tag{2}$$

Here we see how to derive elimination of disjunction $A \vee B$ from instantiation overflow - such derivation is used in the canonical embedding to translate an occurrence of disjunction elimination in a given $\mathbf{IPC}$ proof. The r.h.s. of this equation denotes the elimination of the implication $((A \supset C) \wedge (B \supset C)) \supset C$, implication which is introduced by the inference represented by $\underline{\texttt{io}}(M, A, B, C)$: this is a redex created by the translation, whether or not a redex is present in the given $\mathbf{IPC}$ proof, i.e. whether or not M represents an introduction.

It will turn out that the alternative embedding is more efficient than the original one in avoiding the creation of redexes, because it is based on (1) rather than (2), with instantiation overflow derived from the admissibility of disjunction and absurdity elimination rules, and not the other way around. More precisely, as we will show, the alternative translation of a given $\mathbf{IPC}$ proof is obtained from its canonical translation by reducing redexes that were created at translation time.

Another concern of the present paper is the algorithmic content of the embedding of $\mathbf{IPC}$ into $\mathbf{F_{at}}$. In previous papers on atomic polymorphism (e.g. [2, 3]), the embedding of $\mathbf{IPC}$ into $\mathbf{F_{at}}$ and subsequent studies involving such embedding were done in the natural deduction calculus, with proofs displayed as formula trees. Making use of the Curry-Howard isomorphism, in the present paper we adopt λ-notation not only for the new embedding but also for the original canonical embedding (and instantiation overflow) which are recalled in such notational framework, allowing for more concise proof presentations and for a clearer understanding of the algorithms involved in the processes of proof translation and transformation.

Overview. The paper is structured as follows. In the next section we recall the systems involved in the present study: $\mathbf{IPC}$ and $\mathbf{F_{at}}$. In Section 3 we present the alternative translation of $\mathbf{IPC}$ into $\mathbf{F_{at}}$ We derive instantiation overflow and we prove the soundness of the new translation. In Section 4 we give a detailed analysis of the preservation of proof-reduction steps by the alternative translation. Section 5 recasts the canonical embedding with λ-terms, which allows a precise comparison with the alternative embedding (a supplementary comparison, by means of an example in terms of natural deduction trees, is given as an appendix to the present paper). We finish the paper in Section 6 by discussing whether the "best" translation of $\mathbf{IPC}$ into $\mathbf{F_{at}}$ has already been found.

2 Systems

In the first subsection we present system $\mathbf{IPC}$ while in the second we present atomic system $\mathbf{F}$.

2.1 System IPC

The types/formulas are given by

$$A, B, C \ ::= \ X \mid \ \perp \ \mid A \supset B \mid A \wedge B \mid A \vee B$$

We define $\neg A := A \supset \perp$.

The proof terms M, N, P, Q are inductively generated as follows:

$$\begin{array}{llll} M & ::= & x & \text{(assumption)} \\ & \mid & \lambda x^A.M \mid MN & \text{(implication)} \\ & \mid & \langle M, N\rangle \mid M1 \mid M2 & \text{(conjunction)} \\ & \mid & \mathsf{in}_1(M, A, B) \mid \mathsf{in}_2(N, A, B) \mid \mathsf{case}(M, x^A.P, y^B.Q, C) & \text{(disjunction)} \\ & \mid & \mathsf{abort}(M, A) & \text{(absurdity)} \end{array}$$

We work modulo α-equivalence, in particular we assume the name of the bound variables is always appropriately chosen.

As we will see, the type annotation in the bound variable of binders, in $\mathsf{in}_i(M, A, B)$ and in $\mathsf{abort}(M, A)$ are needed to ensure both the correspondence between proof terms and derivations, and uniqueness of type. The last argument in $\mathsf{case}(M, x^A.P, y^B.Q, C)$ is a type annotation with a different purpose, to be used in the definition of the translation studied in the next section. In any case, such type annotations will often be omitted when possible.

If the components of the pair $\langle P_1, P_2\rangle$ are denoted by long expressions which can be written uniformly on $i = 1, 2$, then we may write this pair as $\langle P_i\rangle_{i=1,2}$.

The typing/inference rules are in Fig. 1. As a logical system, those rules define a natural deduction system for intuitionistic propositional logic. Γ denotes a set of *declarations* $x : A$ such that each variable is declared at most one time in Γ.

Figure 1: Typing/inference rules of **IPC**

$$\frac{}{\Gamma, x : A \vdash x : A}\ Ass$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x^A.M : A \supset B}\ \supset I \qquad \frac{\Gamma \vdash M : A \supset B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B}\ \supset E$$

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \langle M, N \rangle : A \wedge B}\ \wedge I \qquad \frac{\Gamma \vdash M : A \wedge B}{\Gamma \vdash M1 : A}\ \wedge E1 \qquad \frac{\Gamma \vdash M : A \wedge B}{\Gamma \vdash M2 : B}\ \wedge E2$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathsf{in}_1(M, A, B) : A \vee B}\ \vee I1 \qquad \frac{\Gamma \vdash N : B}{\Gamma \vdash \mathsf{in}_2(N, A, B) : A \vee B}\ \vee I2$$

$$\frac{\Gamma \vdash M : A \vee B \quad \Gamma, x : A \vdash P : C \quad \Gamma, y : B \vdash Q : C}{\Gamma \vdash \mathsf{case}(M, x^A.P, y^B.Q, C) : C}\ \vee E$$

$$\frac{\Gamma \vdash M : \perp}{\Gamma \vdash \mathsf{abort}(M, A) : A}\ \perp E$$

A proof term M is *typable* if there are Γ and A such that $\Gamma \vdash M : A$ is derivable from the typing rules. The following proposition explains in what sense does a typable proof term represent a unique typing/logical derivation, and does a typable proof term have a unique type.

Proposition 1. *Given Γ and M, if $\Gamma \vdash M : A$ is derivable for some A, then such an A is unique, and the derivation of $\Gamma \vdash M : A$ is unique.*

Proof. By induction on M. For each case of M, one analyzes the corresponding typing rules. The type annotation in the bound variable of binders ensure one applies the induction hypothesis with a determined set of declarations. The type annotations in $\mathsf{in}_i(M, A, B)$ and in $\mathsf{abort}(M, A)$ ensure uniqueness despite the fresh formulas that show up in the conclusion of rules $\vee I$ and $\perp E$. Notice that the type annotation that constitutes the sixth argument of case plays no role in the analysis of rule $\vee E$. $\square$

For the purpose of defining some reduction rules and the translation of proof terms, it is convenient to arrange the syntax of the system in a different way:

$$\begin{array}{rlll}
\text{(Terms)} & M & ::= & V \,|\, \mathcal{E}[M] \\
\text{(Values)} & V & ::= & x \,|\, \lambda x.M \,|\, \langle M, N \rangle \,|\, \mathsf{in}_1(M, A, B) \,|\, \mathsf{in}_2(N, A, B) \\
\text{(Elim. contexts)} & \mathcal{E} & ::= & [_]N \,|\, [_]1 \,|\, [_]2 \\
 & & | & \mathsf{case}([_], x.P, y.Q, C) \,|\, \mathsf{abort}([_], A)
\end{array}$$

A *value* V ranges over terms representing assumptions or introduction inferences. $\mathcal{E}$ stands for an *elimination context*, which is a term representing an elimination inference,

Figure 2: Typing rules for elimination contexts

$$\frac{}{\Gamma|\perp\vdash \mathsf{abort}([\_],A):A} \qquad \frac{}{\Gamma|A_1\wedge A_2\vdash [\_]i:A_i}\ (i=1,2)$$

$$\frac{\Gamma\vdash N:A}{\Gamma|A\supset B\vdash [\_]N:B} \qquad \frac{\Gamma,x:A\vdash P:C \quad \Gamma,y:B\vdash Q:C}{\Gamma|A\vee B\vdash \mathsf{case}([\_],x.P,y.Q,C):C}$$

$$\frac{\Gamma\vdash M:A \quad \Gamma|A\vdash \mathcal{E}:B}{\Gamma\vdash \mathcal{E}[M]:B}$$

Figure 3: Reduction rules

Detour conversion rules:

$$\begin{array}{lrcll}
(\beta_\supset) & (\lambda x.M)N & \to & [N/x]M & \\
(\beta_\wedge) & \langle M_1,M_2\rangle i & \to & M_i & (i=1,2)\\
(\beta_\vee) & \mathsf{case}(\mathsf{in}_i(M),x_1.P_1,x_2.P_2) & \to & [M/x_i]P_i & (i=1,2)
\end{array}$$

Commutative conversion rules for disjunction:

$$(\pi_\bigcirc)\quad \mathcal{E}_\bigcirc[\mathsf{case}(M,x.P,y.Q)] \quad\to\quad \mathsf{case}(M,x.\mathcal{E}_\bigcirc[P],y.\mathcal{E}_\bigcirc[Q]) \qquad (\bigcirc =\supset,\wedge,\vee,\perp)$$

Commutative conversion rules for absurdity:

$$(\varpi_\bigcirc)\qquad \mathcal{E}_\bigcirc[\mathsf{abort}(M)]\to\mathsf{abort}(M) \qquad (\bigcirc=\wedge,\supset,\vee,\perp)$$

η-rules:

$$\begin{array}{lrcll}
(\eta_\supset) & \lambda x.Mx & \to & M & (x\notin M)\\
(\eta_\wedge) & \langle M1,M2\rangle & \to & M & \\
(\eta_\vee) & \mathsf{case}(M,x.\mathsf{in}_1(x),y.\mathsf{in}_2(y)) & \to & M &
\end{array}$$

but with a "hole" in the position of the main premiss. $\mathcal{E}[M]$ denotes the term resulting from filling the hole of $\mathcal{E}$ with M.

In Fig. 2 one finds the typing rules for elimination contexts. In a sequent $\Gamma|A\vdash \mathcal{E}:B$, the type A is the type of the hole of $\mathcal{E}$ and B is the type of the term obtained by filling the hole of $\mathcal{E}$ with a term of type A.

The reduction rules are given in Fig. 3. The detour conversion rules make use of ordinary substitution $[N/x]M$. The commutative conversion rules make use of a specific organization of the definition of elimination contexts:

$$\begin{array}{lcl@{\qquad}lcl@{\qquad}lcl}
\mathcal{E} & ::= & \mathcal{E}_\supset\,|\,\mathcal{E}_\wedge\,|\,\mathcal{E}_\vee\,|\,\mathcal{E}_\perp & \mathcal{E}_\supset & ::= & [_]N & \mathcal{E}_\vee & ::= & \mathsf{case}([_],x.P,y.Q)\\
 & & & \mathcal{E}_\wedge & ::= & [_]1\,|\,[_]2 & \mathcal{E}_\perp & ::= & \mathsf{abort}([_])
\end{array}$$

We let $\beta:=\beta_\supset\cup\beta_\wedge\cup\beta_\vee$ and similarly for η; we let $\pi:=\pi_\supset\cup\pi_\wedge\cup\pi_\vee\cup\pi_\perp$ and similarly for ϖ. Equivalent definitions of π and ϖ are:

$$\begin{array}{lrcl} (\pi) & \mathcal{E}[\mathsf{case}(M, x.P, y.Q)] & \to & \mathsf{case}(M, x.\mathcal{E}[P], y.\mathcal{E}[Q]) \\ (\varpi) & \mathcal{E}[\mathsf{abort}(M)] & \to & \mathsf{abort}(M)\ . \end{array}$$

Given a reduction rule R of $\mathbf{IPC}$, we employ the usual notations concerning reduction relations generated by R: the compatible closure of R is denoted $\to_R$; and $\to_R^+$, $\to_R^*$, $=_R$ denote respectively the transitive closure, the reflexive-transitive closure, and the reflexive-symmetric-transitive closure of $\to_R$. If $R = R_1 \cup R_2$, then we mau omit "$\cup$" in out notation and write $\to_{R_1 R_2}$, etc. The same notations apply to system $\mathbf{F_{at}}$ to be introduced in the next subsection.

Proposition 2. *Let R be a reduction rule of* $\mathbf{IPC}$. *If $\Gamma \vdash M : A$ is derivable and $M \to_R N$ then $\Gamma \vdash N : A$ is derivable.*

Proof. By induction on $M \to_R N$. □

This is the "subject reduction" property, which states that reduction preserves types. The proof shows how to obtain a derivation of $\Gamma \vdash N : A$ from a given derivation of $\Gamma \vdash M : A$ when $M \to_R N$. The interesting case is the base case, corresponding to the reduction rule itself: the derivation of $\Gamma \vdash N : A$ is obtained by the familiar procedures that eliminate a maximal formula, or shorten a segment, etc. We may see the proof of this proposition as defining the proof transformation induced by the reduction rule R.

2.2 System $\mathbf{F_{at}}$

The atomic system $\mathbf{F}$, denoted $\mathbf{F_{at}}$, is the fragment of system $\mathbf{F}$ induced by restricting to atomic instances the elimination inference rule for $\forall$, and the corresponding proof term constructor. We give a precise definition of $\mathbf{F_{at}}$ by saying what changes relatively to $\mathbf{IPC}$.

Regarding formulas, $\perp$ and $A \vee B$ are dropped, and the new form $\forall X.A$ is adopted. The quantifier $\forall X$ binds free occurrences of X, inducing the obvious concept of free occurrence of a type variable in a type. Concerning α-equivalence, we deal with type variables as we deal with term variables, relying on silent α-renaming. We write $X \notin A$ to say that X does not occur free in A; given the silent α-renaming in A, we may assume X does not occur bound in A either. Another novelty is type substitution in types, which we only require in the atomic form $[Y/X]A$, meaning: substitution in A of each free occurrence of X by Y.

Regarding proof terms, the constructions relative to $\perp$ and $A \vee B$ are dropped, and the new forms $\Lambda X.M$ and MX are added. The latter gives rise to $\mathcal{E}_\forall ::= [\_]X$. Types occur in proof terms, not only via MX, but also via the type annotations in λ-abstractions; in particular, this is how type variables may occur free in proof terms. Accordingly, there is the operation of type substitution in proof terms, denoted $[Y/X]M$, defined by recursion on M: the critical equations are $[Y/X](MX) = ([Y/X]M)Y$ and $[Y/X](\lambda x^A.M) = \lambda x^{[Y/X]A}.[Y/X]M$. Again, we write $X \notin M$ to say that X does not occur free in M, which is the same to say X does not occur at all in M, due to the assumed α-renaming of type variables.

Regarding typing rules, those relative to $\perp$ and $A \vee B$ are dropped, and two rules relative to $\forall X.A$ are adopted:

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \Lambda X.M : \forall X.A}\ \forall I \qquad \frac{\Gamma \vdash M : \forall X.A}{\Gamma \vdash MY : [Y/X]A}\ \forall E_{\mathbf{at}}$$

where the proviso for $\forall I$ is: X does not occur free in some type in Γ. The new form of elimination contexts $\mathcal{E}_\forall$ is typed with:

$$\frac{}{\Gamma|\forall X.A \vdash [\_]Y : [Y/X]A}$$

Regarding reduction rules, we drop commutative conversion rules (since they are relative to $\vee$ and $\perp$). What remains are the β and η-rules (but we drop those relative to disjunction). For $\forall$, these are:

$$\begin{array}{llcll} (\beta_\forall) & (\Lambda X.M)Y & \to & [Y/X]M & \\ (\eta_\forall) & \Lambda X.MX & \to & M & \qquad (X \notin M) \end{array}$$

We let $\beta := \beta_\supset \cup \beta_\wedge \cup \beta_\forall$. Similarly for η.

3 Alternative translation

As an alternative to the canonical embedding of $\mathbf{IPC}$ into $\mathbf{F_{at}}$ [2, 3, 4], in this section we introduce another translation $(\cdot)^\circ : \mathbf{IPC} \to \mathbf{F_{at}}$. The *alternative translation* comprises the Russell-Prawitz translation of formulas and a translation of proof-terms (which induces a translation of derivations). In this section we show the soundness of the translation and the derivation of instantiation overflow.

Definition 1. *In* $\mathbf{F_{at}}$*:*

1. $A \underline{\vee} B := \forall X.((A \supset X) \wedge (B \supset X)) \supset X$*, with* $X \notin A, B$*.*
2. $\underline{\perp} := \forall X.X$*.*

We define the Russell-Prawitz translation of formulas. Using the abbreviations just introduced, the definition can be given in a homomorphic fashion:

$$\begin{array}{rcl} X^\circ & = & X \\ \perp^\circ & = & \underline{\perp} \\ (A \supset B)^\circ & = & A^\circ \supset B^\circ \\ (A \wedge B)^\circ & = & A^\circ \wedge B^\circ \\ (A \vee B)^\circ & = & A^\circ \underline{\vee} B^\circ \end{array}$$

The translation of proof terms will rely on the following, crucial definition:

Definition 2. *In* $\mathbf{F_{at}}$*:*

1. *Given* M, A, B*, given* $i \in \{1, 2\}$*, we define*

$$\underline{\mathtt{in}}_i(M, A, B) := \Lambda X.\lambda w^{(A \supset X) \wedge (B \supset X)}.wiM \ \ ,$$

where the bound variable X *is chosen so that* $X \notin M, A, B$*.*

2. *Given M, P, Q, A, B, C, we define $\underline{\mathtt{case}}(M, x^A.P, y^B.Q, C)$ by recursion on C as follows:*

$$\begin{array}{rcl}
\underline{\mathtt{case}}(M, x^A.P, y^B.Q, X) & = & MX\langle \lambda x^A.P, \lambda y^B.Q\rangle \\
\underline{\mathtt{case}}(M, x^A.P, y^B.Q, C_1 \wedge C_2) & = & \langle \underline{\mathtt{case}}(M, x^A.Pi, y^B.Qi, C_i)\rangle_{i=1,2} \\
\underline{\mathtt{case}}(M, x^A.P, y^B.Q, C \supset D) & = & \lambda z^C.\underline{\mathtt{case}}(M, x^A.Pz, y^B.Qz, D) \\
\underline{\mathtt{case}}(M, x^A.P, y^B.Q, \forall X.C) & = & \Lambda X.\underline{\mathtt{case}}(M, x^A.PX, y^B.QX, C)
\end{array}$$

where, in the third clause, the bound variable z is chosen so that $z \neq x$, $z \neq y$ and $z \notin M, P, Q$; and in the fourth clause, the bound variable X is chosen so that $X \notin M, P, Q, A, B$.

3. *Given M, A, we define $\underline{\mathtt{abort}}(M, A)$ by recursion on A as follows:*

$$\begin{array}{rcl}
\underline{\mathtt{abort}}(M, X) & = & MX \\
\underline{\mathtt{abort}}(M, A_1 \wedge A_2) & = & \langle \underline{\mathtt{abort}}(M, A_1), \underline{\mathtt{abort}}(M, A_2)\rangle \\
\underline{\mathtt{abort}}(M, B \supset C) & = & \lambda z^B.\underline{\mathtt{abort}}(M, C) \\
\underline{\mathtt{abort}}(M, \forall X.A) & = & \Lambda X.\underline{\mathtt{abort}}(M, A)
\end{array}$$

where, in the third clause, the bound variable z is chosen so that $z \notin M$; and in the fourth clause, the bound variable X is chosen so that $X \notin M$.

Next we see how $\underline{\mathtt{in}}$, $\underline{\mathtt{case}}$ and $\underline{\mathtt{abort}}$ behave w.r.t. typing, substitution and compatibility. The first lemma states that the inference rules for disjunction and absurdity are admissible in $\mathbf{F_{at}}$.

Lemma 1. *The typing rules in Fig. 4 are admissible in $\mathbf{F_{at}}$.*

Proof. The first rule has a straightforward proof. Each of the remaining two rules is proved by induction on C. The three proofs use admissibility of weakening in $\mathbf{F_{at}}$: if $\Gamma \vdash M : A$ is derivable and $\Gamma \subseteq \Delta$ then $\Delta \vdash M : A$ is derivable.

We do not give more details, we just argue that the proviso of $\forall I$ is satisfied when typing each occurrence of Λ in the definitions of $\underline{\mathtt{in}}$, $\underline{\mathtt{case}}$ and $\underline{\mathtt{abort}}$. Notice that:

Regarding the definition of $\underline{\mathtt{in}}_i(M, A, B)$, in item 1 of Def. 2, and given Γ satisfying the premiss of the first rule in Fig. 4, the bound variable X can be chosen so that, additionally, X does not occur in a type in Γ.

Regarding the fourth clause of the definition of $\underline{\mathtt{case}}(M, x.P, y.Q, C)$, in item 2 of Def. 2, and given Γ satisfying the three premisses of the second rule in Fig. 4, the bound variable X can be chosen so that, additionally, X does not occur in a type in Γ.

Regarding the fourth clause of the definition of $\underline{\mathtt{abort}}(M, A)$, in item 3 of Def. 2, and given Γ satisfying the premiss of the third rule in Fig. 4, the bound variable X can be chosen so that, additionally, X does not occur in a type in Γ. □

Lemma 2. *In $\mathbf{F_{at}}$:*

1. (a) $[N/z]\underline{\mathtt{in}}_i(M, A, B) = \underline{\mathtt{in}}_i([N/z]M, A, B)$.
   (b) $[Y/X]\underline{\mathtt{in}}_i(M, A, B) = \underline{\mathtt{in}}_i([Y/X]M, [Y/X]A, [Y/X]B)$.
2. (a) $[N/z]\underline{\mathtt{case}}(M, x^A.P, y^B.Q, C) = \underline{\mathtt{case}}([N/z]M, x^A.[N/z]P, y^B.[N/z]Q, C)$.

Figure 4: Admissible typing rules of $\mathbf{F_{at}}$

$$\frac{\Gamma \vdash M : A_i}{\Gamma \vdash \underline{\mathtt{in}}_i(M, A_1, A_2) : A_1 \underline{\vee} A_2}\ (i = 1, 2)$$

$$\frac{\Gamma \vdash M : A \underline{\vee} B \quad \Gamma, x : A \vdash P : C \quad \Gamma, y : B \vdash Q : C}{\Gamma \vdash \underline{\mathtt{case}}(M, x^A.P, y^B.Q, C) : C}$$

$$\frac{\Gamma \vdash M : \underline{\bot}}{\Gamma \vdash \underline{\mathtt{abort}}(M, C) : C}$$

Figure 5: Compatibility rules of $\mathbf{F_{at}}$

$$\frac{M \, R \, M'}{\underline{\mathtt{in}}_i(M, A, B) \, R \, \underline{\mathtt{in}}_i(M', A, B)}$$

$$\frac{M \, R \, M'}{\underline{\mathtt{case}}(M, x.P, y.Q, C) \, R \, \underline{\mathtt{case}}(M', x.P, y.Q, C)}$$

$$\frac{P \, R \, P'}{\underline{\mathtt{case}}(M, x.P, y.Q, C) \, R \, \underline{\mathtt{case}}(M, x.P', y.Q, C)}$$

$$\frac{Q \, R \, Q'}{\underline{\mathtt{case}}(M, x.P, y.Q, C) \, R \, \underline{\mathtt{case}}(M, x.P, y.Q', C)}$$

$$\frac{M \, R \, M'}{\underline{\mathtt{abort}}(M, C) \, R \, \underline{\mathtt{abort}}(M', C)}$$

(b) $[Y/X]\underline{\mathtt{case}}(M, x^A.P, y^B.Q, C) =$
$= \underline{\mathtt{case}}([Y/X]M, x^{A'}.[Y/X]P, y^{B'}.[Y/X]Q, C')$,
where $A' = [Y/X]A$, $B' = [Y/X]B$, *and* $C' = [Y/X]C$.

3. (a) $[N/z]\underline{\mathtt{abort}}(M, C) = \underline{\mathtt{abort}}([N/z]M, C)$.
   (b) $[Y/X]\underline{\mathtt{abort}}(M, C) = \underline{\mathtt{abort}}([Y/X]M, [Y/X]C)$.

Proof. The first two items are immediate. Each of the remaining four are proved by induction on C. □

Lemma 3. *Let R be a relation compatible in the proof-terms of $\mathbf{F_{at}}$. Then the compatibility rules in Fig. 5 are admissible in $\mathbf{F_{at}}$.*

Proof. The first rule is immediate. Each of the remaining four rules is proved by induction on C. □

Figure 6: The translation of proof expressions

$$
\begin{array}{rcll}
x^\circ & = & x & \\
(\lambda x^A.M)^\circ & = & \lambda x^{A^\circ}.M^\circ & \\
\langle M, N\rangle^\circ & = & \langle M^\circ, N^\circ\rangle & \\
(\mathsf{in}_i(M, A, B))^\circ & = & \underline{\mathtt{in}}_i(M^\circ, A^\circ, B^\circ) & (i = 1, 2) \\
(\mathcal{E}_{\bigcirc}[M])^\circ & = & \mathcal{E}^\circ_{\bigcirc}[M^\circ] & (\bigcirc = \supset, \wedge) \\
(\mathsf{case}(M, x^A.P, y^B.Q, C))^\circ & = & \underline{\mathtt{case}}(M^\circ, x^{A^\circ}.P^\circ, y^{B^\circ}.Q^\circ, C^\circ) & \\
(\mathsf{abort}(M, A))^\circ & = & \underline{\mathtt{abort}}(M^\circ, A^\circ) & \\
 & & & \\
([\_]N)^\circ & = & [\_]N^\circ & \\
([\_]i)^\circ & = & [\_]i &
\end{array}
$$

Due to Definition 2, the translation of proof terms can be given in a purely homomorphic fashion:

Definition 3. *Given $M \in \mathbf{IPC}$, M° is defined by recursion on M as in Fig. 6.*

Notice that $(MN)^\circ = M^\circ N^\circ$ and $(Mi)^\circ = M^\circ i$. Also observe the use of the type information provided by the last argument in $\mathsf{case}(M, x^A.P, y^B.Q, C)$: from C we determine the argument C° required by $\underline{\mathtt{case}}$.

Given Γ in $\mathbf{IPC}$, let Γ° denote $\{x : A^\circ | x : A \in \Gamma\}$. The next result means that the alternative embedding works well at the level of provability.

Proposition 3 (Type soundness)**.** *If $\Gamma \vdash M : A$ in $\mathbf{IPC}$, then $\Gamma^\circ \vdash M^\circ : A^\circ$ in $\mathbf{F_{at}}$.*

Proof. By induction on $\Gamma \vdash M : A$, using Lemma 1. $\square$

The constructive content of the proof of this lemma is the proof transformation induced by the proof-term mapping.

What is the role of instantiation overflow in the alternative embedding we just proved? The third rule in Fig. 4 is already the example of instantiation overflow relative to the definition of absurdity in $\mathbf{F_{at}}$. The second rule gives the other example, relative to disjunction, which has been discussed in the introduction of this paper.

Corollary 1 (Instantiation overflow)**.** *Let C be an arbitrary type in $\mathbf{F_{at}}$, and let*

$$\underline{\mathtt{io}}(M, A, B, C) := \lambda z^{(A \supset C) \wedge (B \supset C)}.\underline{\mathtt{case}}(M, x^A.z1x, y^B.z2y, C) \ .$$

The following typing rule is admissible in $\mathbf{F_{at}}$:

$$\frac{\Gamma \vdash M : A \underline{\vee} B}{\Gamma \vdash \underline{\mathtt{io}}(M, A, B, C) : ((A \supset C) \wedge (B \supset C)) \supset C}$$

Proof. Follows immediately from admissibility of the second rule in Fig. 4. $\square$

So instantiation overflow, like the alternative embedding of $\mathbf{IPC}$ into $\mathbf{F_{at}}$, is a consequence of the admissibility of the elimination inference rules for absurdity and disjunction.

4 Analysis of the alternative translation

This section analyzes how the alternative embedding maps proof-reduction steps, leading to Theorem 1. There is a first lemma about the commutation of the embedding with substitution, and a long sequence of lemmas, from Lemma 5 to Lemma 12, about the admissibility in $\mathbf{F_{at}}$ of the reduction rules relative to $\vee$ and $\perp$, that is, the respective β-, η-, and commutative rules. Proofs are given in considerable detail, to allow a later analysis of administrative redexes.

Lemma 4. $[N^\circ/x]M^\circ = ([N/x]M)^\circ$.

Proof. By induction on M. All cases follow by definitions and IH, except the cases $M = \mathsf{in}_i(M_0, A, B)$, $M = \mathsf{case}(M_0, y_1.P_1, y_2.P_2, C)$, and $M = \mathsf{abort}(M_0, A)$, which also require respectively items 1(a), 2(a), and 3(a) of Lemma 2. We just show one of these cases.

Case $M = \mathsf{case}(M_0, y_1.P_1, y_2.P_2, C)$.

$$\begin{array}{rll} & LHS & \\ = & [N^\circ/x]\underline{\mathtt{case}}(M_0^\circ, y_1.P_1^\circ, y_2.P_2^\circ, C^\circ) & \text{((by def. of } (\cdot)^\circ\text{))} \\ = & \underline{\mathtt{case}}([N^\circ/x]M_0^\circ, y_1.[N^\circ/x]P_1^\circ, y_2.[N^\circ/x]P_2^\circ, C^\circ) & \text{(by item 2.(a) of Lemma 2)} \\ = & \underline{\mathtt{case}}(([N/x]M_0)^\circ, y_1.([N/x]P_1)^\circ, y_2.([N/x]P_2)^\circ, C^\circ) & \text{(by IH)} \\ = & RHS & \text{(by defs. of } (\cdot)^\circ \text{ and subst.)} \end{array}$$

$\square$

Lemma 5 (Admissible $\beta_\vee$). *In* $\mathbf{F_{at}}$*:* $\underline{\mathtt{case}}(\underline{\mathtt{in}}_i(N), x_1.P_1, x_2.P_2, C) \to^+_{\beta\eta} [N/x_i]P_i$.

Proof. By induction on C. In each case, the first equality in the calculation is justified by the definition of $\underline{\mathtt{case}}$. Additionally, the first equality of case $C = Y$ uses the definition of $\underline{\mathtt{in}}_i(N)$.

Case $C = Y$.

$$\begin{array}{rcl} LHS & = & (\Lambda X.\lambda z^{(A\supset X)\wedge(B\supset X)}.ziN)Y\langle\lambda x_1.P_1, \lambda x_2.P_2\rangle \\ & \to_{\beta_\forall} & (\lambda z^{(A\supset Y)\wedge(B\supset Y)}.ziN)\langle\lambda x_1.P_1, \lambda x_2.P_2\rangle \\ & \to_{\beta_\supset} & \langle\lambda x_1.P_1, \lambda x_2.P_2\rangle iN \\ & \to_{\beta_\wedge} & (\lambda x_i.P_i)N \\ & \to_{\beta_\supset} & [N/x_i]P_i \end{array}$$

Case $C = D_1 \supset D_2$.

$$\begin{array}{rcll} LHS & = & \lambda z.\underline{\mathtt{case}}(\underline{\mathtt{in}}_i(N), x_1.P_1 z, x_2.P_2 z, D_2) & \\ & \to^+ & \lambda z.[N/x_i](P_i z) & \text{(by IH)} \\ & = & \lambda z.([N/x_i]P_i)z & \\ & \to_{\eta_\supset} & [N/x_i]P_i & \end{array}$$

Case $C = D_1 \wedge D_2$. Similar, uses $\eta_\wedge$.

Case $C = \forall Y.D$.

$$\begin{array}{rcll} LHS & = & \Lambda Y.\underline{\mathtt{case}}(\underline{\mathtt{in}}_i(N), x_1.P_1 Y, x_2.P_2 Y, D) & \\ & \to^+ & \Lambda Y.[N/x_i](P_i Y) & \text{(by IH)} \\ & = & \Lambda Y.([N/x_i]P_i)Y & \\ & \to_{\eta_\forall} & [N/x_i]P_i & \end{array}$$

□

Lemma 6 (Admissible $\eta_\vee$). *In* $\mathbf{F_{at}}$*:*

$$\underline{\texttt{case}}(M, x^A.\underline{\texttt{in}}_1(x, A, B), y^B.\underline{\texttt{in}}_2(y, A, B), A\underline{\vee}B) \to^+_{\beta\eta} M \quad .$$

Proof.

$$\begin{array}{ll}
 & LHS \\
= & \Lambda X.\underline{\texttt{case}}(M, x.(\Lambda Y.\lambda z.z1x)X, y.(\Lambda Y.\lambda z.z2y)X, ((A \supset X) \wedge (B \supset X)) \supset X) \\
\to^2_{\beta_\forall} & \Lambda X.\underline{\texttt{case}}(M, x.\lambda z.z1x, y.\lambda z.z2y, ((A \supset X) \wedge (B \supset X)) \supset X) \\
= & \Lambda X.\lambda w.\underline{\texttt{case}}(M, x.(\lambda z.z1x)w, y.(\lambda z.z2y)w, X) \\
\to^2_{\beta_\supset} & \Lambda X.\lambda w.\underline{\texttt{case}}(M, x.w1x, y.w2y, X) \\
= & \Lambda X.\lambda w.MX\langle \lambda x.w1x, \lambda y.w2y\rangle \\
\to^2_{\eta_\supset} & \Lambda X.\lambda w.MX\langle w1, w2\rangle \\
\to_{\eta_\wedge} & \Lambda X.\lambda w.MXw \\
\to_{\eta_\supset} & \Lambda X.MX \\
\to_{\eta_\forall} & M
\end{array}$$

□

Lemma 7 (Admissible $\pi_{\bigcirc}$, for $\bigcirc = \supset, \wedge, \forall$). *In* $\mathbf{F_{at}}$*:*

1. $(\underline{\texttt{case}}(M, x.P, y.Q, C \supset D))N \to_{\beta_\supset} \underline{\texttt{case}}(M, x.PN, y.QN, D)$.
2. $\underline{\texttt{case}}(M, x.P, y.Q, C_1 \wedge C_2)i \to_{\beta_\wedge} \underline{\texttt{case}}(M, x.Pi, y.Qi, C_i)$.
3. $(\underline{\texttt{case}}(M, x^A.P, y^B.Q, \forall X.C))Y \to_{\beta_\forall} \underline{\texttt{case}}(M, x^A.PY, y^B.QY, [Y/X]C)$.

Proof. Proof of 1.

$$\begin{array}{lll}
 & LHS & \\
= & (\lambda z.\underline{\texttt{case}}(M, x.Pz, y.Qz, D))N & \text{(by def. of } \underline{\texttt{case}}\text{)} \\
\to_{\beta_\supset} & [N/z]\underline{\texttt{case}}(M, x.Pz, y.Qz, D) & \\
= & \underline{\texttt{case}}([N/z]M, x.[N/z](Pz), y.[N/z](Qz), D) & \text{(by item 2.(a) of Lemma 2)} \\
= & RHS & \text{(since } z \notin M, P, Q\text{)}
\end{array}$$

Proof of 2.

$$\begin{array}{rcll}
LHS & = & \langle \underline{\texttt{case}}(M, x.Pj, y.Qj, C_j)\rangle_{j=1,2}i & \text{(by def. of } \underline{\texttt{case}}\text{)} \\
 & \to_{\beta_\wedge} & RHS &
\end{array}$$

Proof of 3.

$$\begin{array}{lll}
 & LHS & \\
= & (\Lambda X.\underline{\texttt{case}}(M, x^A.PX, y^B.QX, C))Y & \text{(by def. of } \underline{\texttt{case}}\text{)} \\
\to_{\beta_\forall} & [Y/X]\underline{\texttt{case}}(M, x^A.PX, y^B.QX, C) & \\
= & \underline{\texttt{case}}(M, x^A.[Y/X](PX), y^B.[Y/X](QX), [Y/X]C) & (*) \\
= & RHS & \text{(since } X \notin P, Q\text{)}
\end{array}$$

where the justification for the equality $(*)$ is item 2.(b) of Lemma 2 and $X \notin M, A, B$.

□

Lemma 8 (Admissible $\varpi_{\bigcirc}$, for $\bigcirc = \supset, \wedge, \forall$). *In* $\mathbf{F_{at}}$*:*

1. $(\underline{\texttt{abort}}(M, A \supset B))N \rightarrow_{\beta_\supset} \underline{\texttt{abort}}(M, B)$.
2. $\underline{\texttt{abort}}(M, A_1 \wedge A_2)i \rightarrow_{\beta_\wedge} \underline{\texttt{abort}}(M, A_i)$, $i = 1, 2$.
3. $(\underline{\texttt{abort}}(M, \forall X.A))Y \rightarrow_{\beta_\forall} \underline{\texttt{abort}}(M, [Y/X]A)$.

Proof. Proof of 1.

$$\begin{array}{rcll} LHS & = & (\lambda z^A.\underline{\texttt{abort}}(M, B))N & \text{(by def. of } \underline{\texttt{abort}}\text{)} \\ & \rightarrow_{\beta_\supset} & [N/z]\underline{\texttt{abort}}(M, B) & \\ & = & \underline{\texttt{abort}}([N/z]M, B) & \text{(by item 3. (a) of Lemma 2)} \\ & = & RHS & \text{(since } z \notin M\text{)} \end{array}$$

Proof of 2.

$$\begin{array}{rcll} LHS & = & \langle \underline{\texttt{abort}}(M, A_1), \underline{\texttt{abort}}(M, A_2)\rangle i & \text{(by def. of } \underline{\texttt{abort}}\text{)} \\ & \rightarrow_{\beta_\wedge} & RHS & \end{array}$$

Proof of 3.

$$\begin{array}{rcll} LHS & = & (\Lambda X.\underline{\texttt{abort}}(M, A))Y & \text{(by def. of } \underline{\texttt{abort}}\text{)} \\ & \rightarrow_{\beta_\forall} & [Y/X]\underline{\texttt{abort}}(M, A) & \\ & = & \underline{\texttt{abort}}([Y/X]M, [Y/X]A) & \text{(by item 3. (b) of Lemma 2)} \\ & = & RHS & \text{(since } X \notin M\text{)} \end{array}$$

□

Lemma 9 (Admissible $\varpi_\vee$). *In* $\mathbf{F_{at}}$*:*

$$\underline{\texttt{case}}(\underline{\texttt{abort}}(M, A\underline{\vee}B), x.P, y.Q, C) \rightarrow^+_\beta \underline{\texttt{abort}}(M, C) \quad .$$

Proof. By induction on C.
Case $C = X$.

$$\begin{array}{rcll} LHS & = & \underline{\texttt{abort}}(M, A\underline{\vee}B)X\langle \lambda x.P, \lambda y.Q\rangle & \text{(by def. of } \underline{\texttt{case}}\text{)} \\ & = & (\Lambda Y \lambda z^{(A\supset Y)\wedge(B\supset Y)}.MY)X\langle \lambda x.P, \lambda y.Q\rangle & \text{(by def. of } \underline{\texttt{abort}}\text{)} \\ & \rightarrow_{\beta_\forall} & (\lambda z^{(A\supset X)\wedge(B\supset X)}.MX)\langle \lambda x.P, \lambda y.Q\rangle & \text{(since } Y \notin M, A, B\text{)} \\ & \rightarrow_{\beta_\supset} & MX & \text{(since } z \notin M\text{)} \\ & = & RHS & \text{(by def. of } \underline{\texttt{abort}}\text{)} \end{array}$$

Case $C = C_1 \supset C_2$.

$$\begin{array}{rcll} LHS & = & \lambda z^{C_1}.\underline{\texttt{case}}(\underline{\texttt{abort}}(M, A\underline{\vee}B), x.Pz, y.Qz, C_2) & \text{(by def. of } \underline{\texttt{case}}\text{)} \\ & \rightarrow^+_\beta & \lambda z^{C_1}.\underline{\texttt{abort}}(M, C_2) & \text{(by IH)} \\ & = & RHS & \text{(by def. of } \underline{\texttt{abort}}\text{)} \end{array}$$

Case $C = C_1 \wedge C_2$.

$$\begin{array}{rcll} LHS & = & \langle \underline{\texttt{case}}(\underline{\texttt{abort}}(M, A\underline{\vee}B), x.Pi, y.Qi, C_i)\rangle_{i=1,2} & \text{(by def. of } \underline{\texttt{case}}\text{)} \\ & \rightarrow^+_\beta & \langle \underline{\texttt{abort}}(M, C_1), \underline{\texttt{abort}}(M, C_2)\rangle & \text{(by IH twice)} \\ & = & RHS & \text{(by def. of } \underline{\texttt{abort}}\text{)} \end{array}$$

Case $C = \forall Y.D$.

$$\begin{array}{lcll} LHS & = & \Lambda Y.\underline{\mathtt{case}}(\underline{\mathtt{abort}}(M, A\underline{\vee}B), x.PY, y.QY, D) & \text{(by def. of } \underline{\mathtt{case}}\text{)} \\ & \to_{\beta}^{+} & \Lambda Y.\underline{\mathtt{abort}}(M, D) & \text{(by IH)} \\ & = & RHS & \text{(by def. of } \underline{\mathtt{abort}}\text{)} \end{array}$$

□

Lemma 10 (Admissible $\varpi_{\perp}$). *In* $\mathbf{F_{at}}$*:*

$$\underline{\mathtt{abort}}(\underline{\mathtt{abort}}(M, \underline{\perp}), A) \to_{\beta_\forall}^{+} \underline{\mathtt{abort}}(M, A) \ .$$

Proof. By induction on A.

Case $A = Y$.

$$\begin{array}{lcll} LHS & = & (\Lambda X.MX)Y & \text{(by def. of } \underline{\mathtt{abort}}\text{)} \\ & \to_{\beta_\forall} & MY & \text{(since } X \notin M\text{)} \\ & = & RHS & \text{(by def. of } \underline{\mathtt{abort}}\text{)} \end{array}$$

Case $A = B \supset C$.

$$\begin{array}{lcll} LHS & = & \lambda z^B.\underline{\mathtt{abort}}(\underline{\mathtt{abort}}(M, \underline{\perp}), C) & \text{(by def. of } \underline{\mathtt{abort}}\text{)} \\ & \to_{\beta_\forall}^{+} & \lambda z^B.\underline{\mathtt{abort}}(M, C) & \text{(by IH)} \\ & = & RHS & \text{(by def. of } \underline{\mathtt{abort}}\text{)} \end{array}$$

Cases $A = B_1 \wedge B_2$ and $A = \forall Y.B$ follow similarly by IH and definition of $\underline{\mathtt{abort}}$. Notice that case $A = B_1 \wedge B_2$ calls the IH twice, and this explains that the lemma is stated with $\to_{\beta_\forall}^{+}$ rather than $\to_{\beta_\forall}$. □

Now we see that the remaining reduction rules of **IPC** hold in $\mathbf{F_{at}}$ as admissible equalities.

Lemma 11 (Admissible $\pi_{\vee}$-equality). *In* $\mathbf{F_{at}}$*:*

$$\begin{array}{l} \underline{\mathtt{case}}(\underline{\mathtt{case}}(M, x_1^{A_1}.P_1, x_2^{A_2}.P_2, B_1\underline{\vee}B_2), y_1^{B_1}.Q_1, y_2^{B_2}.Q_2, C) =_{\beta} \\ \quad \underline{\mathtt{case}}(M, x_1^{A_1}.\underline{\mathtt{case}}(P_1, y_1^{B_1}.Q_1, y_2.Q_2, C), x_2^{A_2}.\underline{\mathtt{case}}(P_2, y_1^{B_1}.Q_1, y_2^{B_2}.Q_2, C), C). \end{array}$$

Proof. By induction on C. The type annotations in bound variables will be omitted after the base case.

Case $C = Y$. The LHS term is, by definition of $\underline{\mathtt{case}}$,

$$(\Lambda X.\lambda w^{(B_1\supset X)\wedge(B_2\supset X)}.MX\langle\lambda x_1^{A_1}.P_1Xw, \lambda x_2^{A_2}.P_2Xw\rangle)Y\langle\lambda y_1^{B_1}.Q_1, \lambda y_2^{B_2}.Q_2\rangle \ ,$$

which, after one $\beta_\forall$-reduction step, becomes

$$(\lambda w^{(B_1\supset Y)\wedge(B_2\supset Y)}.MY\langle\lambda x_1^{A_1}.P_1Yw, \lambda x_2^{A_2}.P_2Yw\rangle)\langle\lambda y_1^{B_1}.Q_1, \lambda y_2^{B_2}.Q_2\rangle \ ,$$

because $X \notin M, P_1, P_2, A_1, A_2, B_1, B_2$. This term, in turn, yields, after one $\beta_\supset$-reduction step,

$$MY\langle\lambda x_1^{A_1}.P_1Y\langle\lambda y_1.Q_1, \lambda y_2.Q_2\rangle, \lambda x_2^{A_2}.P_2Y\langle\lambda y_1^{B_1}.Q_1, \lambda y_2^{B_2}.Q_2\rangle\rangle \ .$$

This is the RHS terms, by definition of $\underline{\texttt{case}}$.

Case $C = C_1 \supset C_2$. The LHS term is, by definition of $\underline{\texttt{case}}$,

$$\lambda z^{C_1}.\underline{\texttt{case}}(\underline{\texttt{case}}(M, x_1.P_1, x_2.P_2, B_1 \underline{\vee} B_2), y_1.Q_1 z, y_2.Q_2 z, C_2) \ ,$$

which, by IH, is β-equal to

$$\lambda z^{C_1}.\underline{\texttt{case}}(M, x_1.\underline{\texttt{case}}(P_1, y_1.Q_1 z, y_2.Q_2 z, C_2), x_2.\underline{\texttt{case}}(P_2, y_1.Q_1 z, y_2.Q_2 z, C_2), C_2) \ . \tag{3}$$

On the other hand, the RHS term is, by definition of $\underline{\texttt{case}}$,

$$\lambda z^{C_1}.\underline{\texttt{case}}(M, x_1.(\underline{\texttt{case}}(P_1, y_1.Q_1, y_2.Q_2, C))z, x_2.(\underline{\texttt{case}}(P_2, y_1.Q_1, y_2.Q_2, C))z, C_2) \ ,$$

which, after two $\beta_{\supset}$-reduction steps (in the "wrong" direction), yields the term (3). These $\beta_{\supset}$-reduction steps are justified by item 1 of Lemma 7.

Case $C = C_1 \wedge C_2$. The LHS terms is, by definition of $\underline{\texttt{case}}$,

$$\langle \underline{\texttt{case}}(\underline{\texttt{case}}(M, x_1.P_1, x_2.P_2, B_1 \underline{\vee} B_2), y_1.Q_1 i, y_2.Q_2 i, C_i) \rangle_{i=1,2} \ ,$$

which, by application of IH twice, is β-equal to

$$\langle \underline{\texttt{case}}(M, x_1.\underline{\texttt{case}}(P_1, y_1.Q_1 i, y_2.Q_2 i, C_i), x_2.\underline{\texttt{case}}(P_2, y_1.Q_1 i, y_2.Q_2 i, C_i), C_i) \rangle_{i=1,2} \ . \tag{4}$$

On the other hand, the RHS term is, by definition of $\underline{\texttt{case}}$,

$$\langle \underline{\texttt{case}}(M, x_1.\underline{\texttt{case}}(P_1, y_1.Q_1, y_2.Q_2, C)i, x_2.\underline{\texttt{case}}(P_2, y_1.Q_1, y_2.Q_2, C)i, C_i) \rangle_{i=1,2} \ ,$$

which, after four $\beta_{\wedge}$-reduction steps (in the "wrong" direction), yields the term (4). These $\beta_{\wedge}$-reduction steps are justified by item 2 of Lemma 7.

Case $C = \forall Y.D$. The LHS term is, by definition of $\underline{\texttt{case}}$,

$$\Lambda Y.\underline{\texttt{case}}(\underline{\texttt{case}}(M, x_1.P_1, x_2.P_2, B_1 \underline{\vee} B_2), y_1.Q_1 Y, y_2.Q_2 Y, D)$$

which, by application of IH, is β-equal to

$$\Lambda Y.\underline{\texttt{case}}(M, x_1.\underline{\texttt{case}}(P_1, y_1.Q_1 Y, y_2.Q_2 Y, D), x_2.\underline{\texttt{case}}(P_2, y_1.Q_1 Y, y_2.Q_2 Y, D), D) \ . \tag{5}$$

On the other hand, the RHS term is, by definition of $\underline{\texttt{case}}$,

$$\Lambda Y.\underline{\texttt{case}}(M, x_1.(\underline{\texttt{case}}(P_1, y_1.Q_1, y_2.Q_2, C))Y, x_2.(\underline{\texttt{case}}(P_2, y_1.Q_1, y_2.Q_2, C))Y, D) \ ,$$

which, after two $\beta_{\forall}$-reduction steps (in the "wrong" direction), yields the term (5). These $\beta_{\forall}$-reduction steps are justified by item 3 of Lemma 7. $\square$

Lemma 12 (Admissible $\pi_{\perp}$-equality)**.** *In* $\mathbf{F_{at}}$*:*

$$\underline{\texttt{abort}}(\underline{\texttt{case}}(M, x^A.P, y^B.Q, \underline{\perp}), C) =_{\beta} \underline{\texttt{case}}(M, x^A.\underline{\texttt{abort}}(P, C), y^B.\underline{\texttt{abort}}(Q, C), C) \ .$$

Proof. By induction on C. The proof has the same pattern as that of the previous lemma. The inductive cases generate β-reduction steps in the "wrong" direction, justified by Lemma 8. The type annotations in bound variables will be omitted after the base case.

Case $C = Y$.

$$
\begin{array}{lll}
LHS & = & \underline{\texttt{case}}(M, x^A.P, y^B.Q, \underline{\perp})Y & \text{(by def. of } \underline{\texttt{abort}}\text{)} \\
& = & (\Lambda X.MX\langle \lambda x^A.PX, \lambda y^B.QX\rangle)Y & \text{(by def. of } \underline{\texttt{case}}\text{)} \\
& \to_{\beta_\forall} & MY\langle \lambda x^A.PY, \lambda y^B.QY\rangle & \text{(since } X \notin M, P, Q, A, B\text{)} \\
& = & MY\langle \lambda x^A.\underline{\texttt{abort}}(P, Y), \lambda y^B.\underline{\texttt{abort}}(Q, Y)\rangle & \text{(by def. of } \underline{\texttt{abort}}\text{)} \\
& = & RHS & \text{(by def. of } \underline{\texttt{case}}\text{)}
\end{array}
$$

Case $C = C_1 \supset C_2$. The LHS term is, by definition of $\underline{\texttt{abort}}$,

$$\lambda z^{C_1}.\underline{\texttt{abort}}(\underline{\texttt{case}}(M, x.P, y.Q, \underline{\perp}), C_2) \ \ ,$$

which, by IH, is β-equal to

$$\lambda z^{C_1}.\underline{\texttt{case}}(M, x.\underline{\texttt{abort}}(P, C_2), y.\underline{\texttt{abort}}(Q, C_2), C_2) \ \ . \tag{6}$$

On the other hand, the RHS term is, by definition of $\underline{\texttt{case}}$,

$$\lambda z^{C_1}.\underline{\texttt{case}}(M, x.\underline{\texttt{abort}}(P, C_1 \supset C_2)z, y.\underline{\texttt{abort}}(Q, C_1 \supset C_2)z, C_2) \ \ ,$$

which $\beta_\supset$-reduces (in the "wrong direction") to (6). The reduction is justified by item 1 of Lemma 8.

Case $C = C_1 \wedge C_2$. The LHS term, is, by definition of $\underline{\texttt{abort}}$,

$$\langle \underline{\texttt{abort}}(\underline{\texttt{case}}(M, x.P, y.Q, \underline{\perp}), C_i)\rangle_{i=1,2} \ \ ,$$

which, by IH applied twice, is β-equal to

$$\langle \underline{\texttt{case}}(M, x.\underline{\texttt{abort}}(P, C_i), y.\underline{\texttt{abort}}(Q, C_i), C_i)\rangle_{i=1,2} \ \ . \tag{7}$$

On the other hand, the RHS term is, by definition of $\underline{\texttt{case}}$,

$$\langle \underline{\texttt{case}}(M, x.\underline{\texttt{abort}}(P, C_1 \wedge C_2)i, y.\underline{\texttt{abort}}(Q, C_1 \wedge C_2)i, C_i)\rangle_{i=1,2} \ \ ,$$

which $\beta_\wedge$-reduces (in the "wrong direction") to (7). The reduction is justified by item 2 of Lemma 8.

Case $C = \forall Y.D$. The LHS term is, by definition of $\underline{\texttt{abort}}$,

$$\Lambda Y.\underline{\texttt{abort}}(\underline{\texttt{case}}(M, x.P, y.Q, \underline{\perp}), D) \ \ ,$$

which, by IH, is β-equal to

$$\Lambda Y.\underline{\texttt{case}}(M, x.\underline{\texttt{abort}}(P, D), y.\underline{\texttt{abort}}(Q, D), D) \ \ . \tag{8}$$

On the other hand, the RHS term is, by definition of $\underline{\texttt{case}}$,

$$\Lambda Y.\underline{\texttt{case}}(M, x.(\underline{\texttt{abort}}(P, \forall Y.D))Y, y.(\underline{\texttt{abort}}(Q, \forall Y.D))Y, D) \ \ ,$$

which $\beta_\forall$-reduces (in the "wrong direction") to (8). The reduction is justified by item 3 of Lemma 8. $\square$

We are ready to give the full picture of how $(\cdot)^\circ$ maps reduction steps.

Theorem 1. *Let R be a reduction rule of $\mathbf{IPC}$ given in Fig. 3.*

- *Case $R \notin \{\pi_\vee, \pi_\perp\}$: if $M \to_R N$ in $\mathbf{IPC}$, then $M^\circ \to^+ N^\circ$ in $\mathbf{F_{at}}$.*
- *Case $R \in \{\pi_\vee, \pi_\perp\}$: if $M \to_R N$ in $\mathbf{IPC}$, then $M^\circ =_\beta N^\circ$ in $\mathbf{F_{at}}$.*

Proof. By induction on $M \to N$. Let us check the base cases.

Case $R \in \{\beta_\wedge, \eta_\wedge, \eta_\supset\}$. Trivially one has $M^\circ \to_R N^\circ$.

Case $R = \beta_\supset$: Again $M^\circ \to_R N^\circ$, using Lemma 4.

Case $R = \beta_\vee$: By Lemmas 4 and 5.

Case $R = \eta_\vee$: By Lemma 6.

Case $R = \pi_\bigcirc$, $\bigcirc = \supset, \wedge$: By Lemma 7.

Case $R = \varpi_\bigcirc$, $\bigcirc = \supset, \wedge$: By Lemma 8.

Case $R = \varpi_\bigcirc$, $\bigcirc = \vee, \perp$: By Lemmas 9 and 10.

Case $R = \pi_\bigcirc$, $\bigcirc = \vee, \perp$: By Lemmas 11 and 12.

Inductive cases are routine since the relations $\to^+$ and $=_\beta$ are compatible relations (hence enjoy the compatibility rules in Fig. 5).

$\square$

Comments on Theorem 1. Given a reduction step $M \to N$ in $\mathbf{IPC}$, we give an analysis of the reduction steps in $\mathbf{F_{at}}$ between M° and N°, profiting from the detailed proofs given before, and trying to "explain" why Theorem 1 fails to preserve the direction of reduction in some cases. The interesting reduction rules are those *relative to disjunction or absurdity*, that is, $\beta_\vee$, $\eta_\vee$, and the commutative rules $\pi_\bigcirc$ and $\varpi_\bigcirc$. We have to go back to Def. 2, and make four observations.

First observation. As seen in its definition (item 1 of Def. 2), the term $\underline{\mathtt{in}}_i(M, A, B)$ is expecting some data: an atomic type X and a pair of type $(A \supset X) \wedge (B \supset X)$. This data is provided in the base case of the definition of $\underline{\mathtt{case}}(M, x.P, y.Q, C)$, that is, the case $C = X$: the atomic type X and the pair $\langle \lambda x.P, \lambda y.Q \rangle$ are ready to be "passed" to M. Sometimes, the data request meets the data provision: we see this happening in the first reduction steps either of the base case of the proof of Lemma 5, or the proof of Lemma 6.

Second observation. The definition of $\underline{\mathtt{case}}(M, x.P, y.Q, C)$ generates a tree of recursive calls homomorphic to the syntactic tree of type C, whose leaves correspond to the occurrences of atomic types in C. In each such leave, relative to atomic type X, say, one finds $MX\langle \lambda x.P', \lambda y.Q' \rangle$, where P' and Q' are respectively P and Q applied to the same sequence of formal parameters and projection symbols, leading from type C to type X; and that sequence is dictated by the sequence of abstractions and pairings corresponding to the path in the tree from the root to such a leave. This has the flavor of η-expansion, and indeed one sees two situations where one has to recover from such "η-expansion" by doing η-reduction: in the non-atomic cases of the proof of Lemma 5, and in the proof of Lemma 6.

Third observation. The term $\underline{\mathtt{case}}(M, x.P, y.Q, C)$ (resp. $\underline{\mathtt{abort}}(M, A)$) is used to translate an elimination inference in $\mathbf{IPC}$; but, when C (resp. A) is not atomic, the term does not represent an elimination inference in $\mathbf{F_{at}}$, as it begin with an abstraction or is a pair. This is useful most of the time: that is how commutative reductions in $\mathbf{IPC}$ are turned into β-reductions in $\mathbf{F_{at}}$ - recall the proofs of Lemmas 7, 8, 9, and 10. Even in the proofs of Lemmas 11 and 12 we see that the LHS term starts doing β-reduction in the "correct" direction. However, there is a risk in such a translation of an elimination by an introduction - as we see in the next observation.

Fourth observation. In the second, third and fourth clauses of the definition of $\underline{\texttt{case}}(M, x.P, y.Q, C)$, the expressions Pi, Pz and PX are redexes if P is a pair or abstraction (and similarly for Q). This happens (1) when P is the translation of an introduction inference in $\mathbf{IPC}$ - see for instance the LHS term in the proof of Lemma 6; (2) when P or Q are the translation of some inference in $\mathbf{IPC}$ eliminating disjunction or absurdity, that is, P or Q are some $\underline{\texttt{case}}$ or $\underline{\texttt{abort}}$ - see the RHS terms of the non-atomic cases of the proofs of Lemmas 11 and 12: in such cases a redex is seen in $\underline{\texttt{case}}(M, x.P, y.Q, C)$ that corresponds to no redex in the source $\mathbf{IPC}$-proof. We call *administrative* such redexes, which are created by the translation, "at translation time". As it happens, the reduction of administrative redex is often needed, as seen both in the proof of Lemma 6, and in the non-atomic cases of the proofs of Lemmas 11 and 12 - and the reduction of the latter is the only reduction that goes in the "wrong" direction in the entire proof of Theorem 1.

Summarizing, if R is a reduction rule of $\mathbf{IPC}$ relative to disjunction or absurdity, a reduction step $M \to_R N$ gives rise to the following diagram, where double-headed arrows denote 0, 1 or more reduction steps:

$$\begin{tikzcd}
M \arrow[r, mapsto] \arrow[d, "R"'] & M^\circ \arrow[rr, two heads, "admin"] & & \bullet \arrow[d, two heads, "\beta\eta"] \\
N \arrow[r, mapsto] & N^\circ \arrow[rr, two heads, "admin"] & & \bullet
\end{tikzcd}$$

5 The canonical translation

In this section, we compare the alternative embedding of $\mathbf{IPC}$ into $\mathbf{F_{at}}$ we proposed in Section 3 with the original, canonical embedding. This requires to present the latter in λ-notation, which is in itself interesting, as it works out the algorithmic content of the canonical embedding. (In the appendix to this paper we compare, using natural deduction derivations, the alternative and canonical translations of a certain $\mathbf{IPC}$ proof.)

The canonical embedding is by now described in several publications but always in natural deduction style. See for instance [3, 2] or [5, 4]. In the former references conjunction is interpreted in $\mathbf{F_{at}}$ by the Russell-Prawitz's translation of formulas while in the latter references conjunction is a primitive symbol of $\mathbf{F_{at}}$. Atomic $\mathbf{F}$ was developed with the purpose of avoiding the "bad" connectives of the natural deduction calculus (see the eloquent exposition about the defects of some natural deduction rules in [6] Chapter 10). Being $\wedge$ a "good" connective there is no obstacle in considering it as primitive in $\mathbf{F_{at}}$. On the contrary, recent studies in the canonical translation [5, 4]

Figure 7: Admissible typing rule in $\mathbf{F_{at}}$

$$\frac{\Gamma \vdash M : A\underline{\vee}B}{\Gamma \vdash \underline{\texttt{cio}}(M, A, B, C) : ((A \supset C) \wedge (B \supset C)) \supset C}$$

show that the implementation of the η-conversions into atomic $\mathbf{F}$ or the validity of the Rasiowa-Harrop disjunction property in $\mathbf{F_{at}}$ require conjunction to be primitive in the system. To help the comparison with the alternative translation of the previous sections which considers $\wedge$ as primitive in $\mathbf{F_{at}}$ we take the canonical translation with primitive conjunction in the target system. Also, for the sake of comparability, in the canonical embedding we consider the translation of $A \vee B$ in the form $\forall X.((A \supset X) \wedge (B \supset X)) \supset X$ instead of $\forall X.(A \supset X) \supset ((B \supset X) \supset X)$ (see [5], Final Comment (2)).

The canonical translation of formulas is exactly the one presented in the beginning of Section 3 – Russell-Prawitz's translation. The canonical translation of proofs relies crucially on the phenomenon of instantiation overflow: (i) given in $\mathbf{F_{at}}$ a proof M of $\underline{\perp}$ and an arbitrary formula C, there is a proof in $\mathbf{F_{at}}$ of C obtained from M, which is represented by the previously defined term $\underline{\texttt{abort}}(M, C)$; (ii) given in $\mathbf{F_{at}}$ a proof M of $A\underline{\vee}B$ and an arbitrary formula C, there is a proof in $\mathbf{F_{at}}$ of $((A \supset C) \wedge (B \supset C)) \supset C$ obtained from M, which is represented by $\underline{\texttt{cio}}(M, A, B, C)$, defined as follows:

Definition 4 (Canonical instantiation overflow)**.** *In $\mathbf{F_{at}}$: Given M, A, B, C, we define $\underline{\texttt{cio}}(M, A, B, C)$ by recursion on C as follows:*

$$\begin{array}{rcl}
\underline{\texttt{cio}}(M, A, B, X) & = & MX \\
\underline{\texttt{cio}}(M, A, B, C_1 \wedge C_2) & = & \lambda z.\langle \underline{\texttt{cio}}(M, A, B, C_i)\langle \lambda x^A.z1xi, \lambda y^B.z2yi\rangle\rangle_{i=1,2} \\
\underline{\texttt{cio}}(M, A, B, C_1 \supset C_2) & = & \lambda z.\lambda u^{C_1}.\underline{\texttt{cio}}(M, A, B, C_2)\langle \lambda x^A.z1xu, \lambda y^B.z2yu\rangle \\
\underline{\texttt{cio}}(M, A, B, \forall X.C_1) & = & \lambda z.\Lambda X.\underline{\texttt{cio}}(M, A, B, C_1)\langle \lambda x^A.z1xX, \lambda y^B.z2yX\rangle
\end{array}$$

where, in the second, third and fourth clauses, the bound variables z is chosen so that $z \notin M$; in the fourth clause, the bound variable X is chosen so that $X \notin M, A, B$; and, in the second, third, and fourth clauses, the type of the bound variable z is respectively $(A \supset (C_1 \wedge C_2)) \wedge (B \supset (C_1 \wedge C_2))$, $(A \supset C_1 \supset C_2) \wedge (B \supset C_1 \supset C_2)$, and $(A \supset \forall X.C_1) \wedge (B \supset \forall X.C_1)$.

Lemma 13. *The typing rule in Fig. 7 is admissible in $\mathbf{F_{at}}$.*

Proof. By induction on C. As in the proof of Lemma 1, we rely on admissibility of weakening in $\mathbf{F_{at}}$. We just argue that the proviso of $\forall I$ is satisfied when typing the occurrence of Λ in the fourth clause of the definition of $\underline{\texttt{cio}}(M, A, B, C)$. Notice that, given Γ satisfying the premiss of the rule in Fig. 7, not only $X \notin M, A, B$, but also we may assume X does not occur in a type in Γ. Therefore, X does not occur in a type in $\Gamma, z : (A \supset \forall X.C_1) \wedge (B \supset \forall X.C_1)$. $\square$

The constructive contents of the proof of this lemma is the proof transformation that underlies instantiation overflow as captured in [5, 4]. Def. 4 gives the algorithmic content of such proof transformation.

Lemma 14. *The following compatibility rule is admissible in* $\mathbf{F_{at}}$*:*

$$\frac{M\,R\,M'}{\underline{\mathtt{cio}}(M,A,B,C)\,R\,\underline{\mathtt{cio}}(M',A,B,C)}$$

Definition 5 (Canonical translation)**.** *Given* $M \in \mathbf{IPC}$*, the* canonical translation *of* M*, written* $M^\star$*, is defined by recursion on* M *exactly as* M°*, except for one case, which now reads:*

$$(\mathsf{case}(M, x^A.P, y^B.Q, C))^\star = \underline{\mathtt{cio}}(M^\star, A^\star, B^\star, C^\star)\langle \lambda x^{A^\star}.P^\star, \lambda y^{B^\star}.Q^\star\rangle$$

Lemma 15. *In* $\mathbf{F_{at}}$*:*

$$\underline{\mathtt{cio}}(M,A,B,C)\langle \lambda x^A.P, \lambda y^B.Q\rangle \to_\beta^* \underline{\mathtt{case}}(M, x^A.P, y^B.Q, C)$$

Proof. Fixing M, A and B we prove, by induction on C, that for all terms P and Q we have $\underline{\mathtt{cio}}(M,A,B,C)\langle \lambda x^A.P, \lambda y^B.Q\rangle \to_\beta^+ \underline{\mathtt{case}}(M, x^A.P, y^B.Q, C)$.

Case $C = X$.

$$\begin{array}{rcll} LHS & = & (MX)\langle \lambda x^A.P, \lambda y^B.Q\rangle & \text{(by def. of } \underline{\mathtt{cio}}\text{)} \\ & = & RHS & \text{(by def. of } \underline{\mathtt{case}}\text{)} \end{array}$$

Case $C = C_1 \supset C_2$.

$$\begin{array}{cll} & LHS & \\ = & (\lambda z.\lambda u^{C_1}.\underline{\mathtt{cio}}(M,A,B,C_2)\langle \lambda w^A.z1wu, \lambda r^B.z2ru\rangle)\langle \lambda x^A.P, \lambda y^B.Q\rangle & \text{(a)} \\ \to_\beta^+ & \lambda u^{C_1}.\underline{\mathtt{cio}}(M,A,B,C_2)\langle \lambda w^A.(\lambda x^A.P)wu, \lambda r^B.(\lambda y^B.Q)ru\rangle) & \\ \to_\beta^+ & \lambda u^{C_1}.\underline{\mathtt{cio}}(M,A,B,C_2)\langle \lambda x^A.Pu, \lambda y^B.Qu\rangle & \\ \to_\beta^* & \lambda u^{C_1}.\underline{\mathtt{case}}(M, x^A.Pu, y^B.Qu, C_2) & \text{(b)} \\ = & RHS & \text{(c)} \end{array}$$

Justifications: (a) By definition of $\underline{\mathtt{cio}}$. (b) By IH. (c) By definition of $\underline{\mathtt{case}}$.

Case $C = C_1 \wedge C_2$.

$$\begin{array}{cll} & LHS & \\ = & (\lambda z.\langle \underline{\mathtt{cio}}(M,A,B,C_i)\langle \lambda w^A.z1wi, \lambda r^B.z2ri\rangle\rangle_{i=1,2})\langle \lambda x^A.P, \lambda y^B.Q\rangle & \text{(a)} \\ \to_\beta^+ & \langle \underline{\mathtt{cio}}(M,A,B,C_i)\langle \lambda w^A.(\lambda x^A.P)wi, \lambda r^B.(\lambda y^B.Q)ri\rangle\rangle_{i=1,2} & \\ \to_\beta^+ & \langle \underline{\mathtt{cio}}(M,A,B,C_i)\langle \lambda x^A.Pi, \lambda y^B.Qi\rangle\rangle_{i=1,2} & \\ \to_\beta^* & \langle \underline{\mathtt{case}}(M, x^A.Pi, y^B.Qi, C_i)\rangle_{i=1,2} & \text{(b)} \\ = & RHS & \text{(c)} \end{array}$$

Justifications: (a) By definition of $\underline{\mathtt{cio}}$. (b) By IH twice. (c) By definition of $\underline{\mathtt{case}}$.

Case $C = \forall X C_1$.

$$\begin{array}{cll} & LHS & \\ = & (\lambda z.\Lambda X.\underline{\mathtt{cio}}(M,A,B,C_1)\langle \lambda w^A.z1wX, \lambda r^B.z2rX\rangle)\langle \lambda x^A.P, \lambda y^B.Q\rangle & \text{(a)} \\ \to_\beta^+ & \Lambda X.\underline{\mathtt{cio}}(M,A,B,C_1)\langle \lambda w^A.(\lambda x^A.P)wX, \lambda r^B.(\lambda y^B.Q)rX\rangle) & \\ \to_\beta^+ & \Lambda X.\underline{\mathtt{cio}}(M,A,B,C_1)\langle \lambda x^A.PX, \lambda y^B.QX\rangle & \\ \to_\beta^* & \Lambda X.\underline{\mathtt{case}}(M, x^A.PX, y^B.QX, C_1) & \text{(b)} \\ = & RHS & \text{(c)} \end{array}$$

Justifications: (a) By definition of $\underline{\mathtt{cio}}$. (b) By IH. (c) By definition of $\underline{\mathtt{case}}$. $\square$

Theorem 2 (Canonical vs. alternative translations)**.** *For all $M \in \mathbf{IPC}$, $M^\star \to_\beta^* M^\circ$.*

Proof. The proof is by induction on M. Since $(\cdot)^\star$ and $(\cdot)^\circ$ coincide except for the elimination of disjunction, we just need to prove that

$$(\mathsf{case}(N, x^A.P, y^B.Q, C))^\star \to_\beta^* (\mathsf{case}(N, x^A.P, y^B.Q, C))^\circ$$

given the induction hypotheses $N^\star \to_\beta^* N^\circ$, $P^\star \to_\beta^* P^\circ$ and $Q^\star \to_\beta^* Q^\circ$.

We have

$$\begin{array}{rcll} LHS & = & \underline{\mathtt{cio}}(N^\star, A^\star, B^\star, C^\star)\langle \lambda x^{A^\star}.P^\star, \lambda y^{B^\star}.Q^\star\rangle & \text{(by def. of } (\cdot)^\star) \\ & \to_\beta^* & \underline{\mathtt{case}}(N^\star, x^{A^\star}.P^\star, y^{B^\star}.Q^\star, C^\star) & \text{(by Lemma 15)} \\ & \to_\beta^* & \underline{\mathtt{case}}(N^\circ, x^{A^\circ}.P^\circ, y^{B^\circ}.Q^\circ, C^\circ) & \text{(by IH + Lemma 3)} \\ & = & RHS & \text{(by def. of } (\cdot)^\circ) \end{array}$$

$\square$

Comments on Theorem 2. We argue that the reduction $M^\star \to_\beta^* M^\circ$ stated in Theorem 2 is administrative, in the following sense: it starts by the reduction of an administrative redex in $M^\star$, continues with the immediate reduction of the redexes created by this initial reduction step, and continues, if it continues at all, by picking another administrative redex, that is the "descendant" of some redex already present in the initial term, and repeating this process. Incidentally we observe that, if $M^\star \to_\beta^* M^\circ$, then the size of $M^\star$ is bigger than the size of M°.

We make a preliminary remark. If C is not atomic, $\underline{\mathtt{cio}}(M, A, B, C)$ is an abstraction. This has two consequences. On the one hand, the second (resp. third, fourth) equation in Def. 4 creates a redex, if some C_i (resp. C_2, C_1) is not atomic. Let us refer to such a redex as a *redex created at* $\underline{\mathtt{cio}}(M, A, B, C)$. On the other hand, the definition of $(\cdot)^\star$ creates several redexes whenever it translates an occurrence of $\mathsf{case}(M, x.P, y.Q, C)$ with C non-atomic: not only the redexes created at $\underline{\mathtt{cio}}(M^\star, A^\star, B^\star, C^\star)$ (and possibly at subtypes of $C^\star$), but also the term displayed in Def. 5. All these redexes are administrative.

Let us define

$$\underline{\mathtt{CASE}}(M, x^A.P, y^B.Q, C) := \underline{\mathtt{cio}}(M, A, B, C)\langle \lambda x^A.P, \lambda y^B.Q\rangle \ .$$

This definition recalls (2) in the introduction of the present paper; and reduces the difference between the canonical and the alternative embeddings to the difference between translating case with $\underline{\mathtt{case}}$ or $\underline{\mathtt{CASE}}$. Additionally, Lemma 15 can be stated as $\underline{\mathtt{CASE}}(M, x^A.P, y^B.Q, C) \to_\beta^* \underline{\mathtt{case}}(M, x^A.P, y^B.Q, C)$.

Recall the equations defining $\underline{\mathtt{case}}$ in Def. 2. They should be contrasted with the following reductions, seen to hold simply by inspecting the proof of Lemma 15:

$$\begin{array}{rcl} \underline{\mathtt{CASE}}(M, x.P, y.Q, C \supset D) & \to_\beta^* & \lambda z^C.\underline{\mathtt{CASE}}(M, x.Pz, y.Qz, D) \\ \underline{\mathtt{CASE}}(M, x.P, y.Q, C_1 \wedge C_2) & \to_\beta^* & \langle \underline{\mathtt{CASE}}(M, x.Pi, y.Qi, C_i)\rangle_{i=1,2} \\ \underline{\mathtt{CASE}}(M, x^A.P, y^B.Q, \forall X.C) & \to_\beta^* & \Lambda X.\underline{\mathtt{CASE}}(M, x.PX, y.QX, C) \end{array} \qquad (9)$$

We could have taken (9) as axioms generating the reduction relation stated by Lemma 15; therefore, the same if true of the reduction relation stated by Theorem 2, because the latter is generated by a single call to Lemma 15, in the inductive case relative to $M = \mathsf{case}(N, x^A.P, y^B.Q, C)$ displayed in the proof of Theorem 2. Reduction $M^\star \to_\beta^* M^\circ$ is shown to be administrative by an analysis of reductions (9).

We detail the analysis of the first reduction in (9), the one relative to $C \supset D$. Let $LHS := \underline{\mathtt{CASE}}(M, x.P, y.Q, C \supset D)$ and $RHS := \lambda z^C.\underline{\mathtt{CASE}}(M, x.Pz, y.Qz, D)$. It is evident that the size of LHS is bigger than the size of RHS. Based on the preliminary remark above and its immediate consequences, and on the inspection of the pertinent inductive case in the proof of Lemma 15, we make three additional observations. (1) LHS is a redex and the reduction $LHS \to_\beta^* RHS$ consists in an initial step reducing this redex, followed by the immediate reduction of the redexes created by this initial step. (2) LHS contains the term $\underline{\mathtt{cio}}(M, A, B, C \supset D)$. (3) If D is not atomic, then: (i) LHS contains a redex created at $\underline{\mathtt{cio}}(M, A, B, C \supset D)$; (ii) $\underline{\mathtt{CASE}}(M, x.Pz, y.Qz, D)$ is a redex contained in RHS; (iii) the reduction $LHS \to_\beta^* RHS$ transforms the redex created at $\underline{\mathtt{cio}}(M, A, B, C \supset D)$into the redex $\underline{\mathtt{CASE}}(M, x.Pz, y.Qz, D)$, hence the latter is a "descendant" of the former.

We will omit the similar observations about the second and third reductions in (9), relative to $C_1 \wedge C_2$ and $\forall X.C$.

We end this discussion by completing the diagram at the end of Section 4, induced by a reduction step $M \to_R N$ in $\mathbf{IPC}$, with R a reduction rule relative to disjunction or absurdity.

$$\begin{tikzcd}
M \arrow[r, mapsto] \arrow[d, "R"'] & M^\star \arrow[r, two heads, "admin"] & M^\circ \arrow[r, two heads, "admin"] & \bullet \arrow[d, two heads, "\beta\eta"] \\
N \arrow[r, mapsto] & N^\star \arrow[r, two heads, "admin"] & N^\circ \arrow[r, two heads, "admin"] & \bullet
\end{tikzcd}$$

6 Final remarks

In this paper we proposed an alternative embedding of $\mathbf{IPC}$ into atomic system $\mathbf{F}$, based on the admissibility of disjunction and absurdity elimination rules, rather than instantiation overflow, and proved that the alternative embedding works as well as the original one at the levels of provability and preservation of proof reduction. In fact, the alternative embedding preserves $\beta\eta$-conversions and maps commutative conversions to β-equalities, exactly as the original embedding; but the alternative embedding is more economical, as it produces $\mathbf{F_{at}}$ proofs of smaller size and $\mathbf{F_{at}}$ simulations of smaller length. In this sense, we may speak of a "refined" embedding.

Given the existence of an alternative embedding, one cannot view the existence of an embedding of $\mathbf{IPC}$ into atomic system $\mathbf{F}$ as necessarily based on the phenomenon of instantiation overflow, and one immediately questions, not only what is the role of that phenomenon in the embedding, but also (and mainly) whether there is a "truly canonical” translation of $\mathbf{IPC}$ proofs into $\mathbf{F_{at}}$ proofs - as stable as Russell-Prawitz translation at the level of formulas.

Our results go far enough to sketch what an answer to the latter question might be. The key technical tool is that of an “administrative” redex, a redex created by the translation itself. We have proved that the translation of a given $\mathbf{IPC}$ proof by the alternative embedding is obtained from the translation of the same proof by the original embedding through the reduction of administrative redexes which the original translation created (this already explains why the alternative embedding is more economical, and in what sense the alternative embedding is an optimization of the original one). In addition, the alternative embedding also creates administrative redexes. This explains why the alternative embedding falls short of delivering preservation of reduction steps in all cases; but, more important, this suggest how the alternative embedding could be further optimized. In fact, the main suggestion we offer is that the “truly canonical” embedding will be the embedding free from administrative redexes.

So the final question is: is there such a fully optimized, “truly canonical” embedding, which: (i) can be defined by recursion on the syntax of the given $\mathbf{IPC}$ proof; (ii) by virtue of being free from administrative redexes, delivers preservation of reduction steps in all cases? We do not have an answer. But, having developed the original and the alternative embeddings with λ-terms, we can now recognize the problem as a problem of program optimization, and conjecture that it can be solved by techniques like reduction “on the fly” (at compile time) that have been successfully employed since long in the study of other program transformations [7].

A Appendix

Since most of the previous work on $\mathbf{F_{at}}$ was written in the natural deduction calculus, we illustrate, for curiosity, the difference between the canonical and the alternative translation in tree-style derivation, showing how a concrete proof in $\mathbf{IPC}$ is translated into $\mathbf{F_{at}}$ via both the above embeddings.

Let $\mathcal{D}$ be the following derivation of $(\neg Z)\vee W \vdash Z\supset W$ in $\mathbf{IPC}$:

$$\dfrac{(\neg Z)\vee W \qquad \dfrac{\dfrac{\dfrac{[Z\supset\bot]^v \qquad [Z]^u}{\bot}\supset E}{W}\bot E}{Z\supset W}\supset I_u \qquad \dfrac{[W]^v}{Z\supset W}\supset I}{Z\supset W}\vee E_v$$

Via the canonical translation, we obtain the following derivation $\mathcal{D}^\star$ in $\mathbf{F_{at}}$:

$$\dfrac{\dfrac{\forall X.(((Z\supset\forall Y.Y)\supset X)\wedge(W\supset X))\supset X}{(((Z\supset\forall Y.Y)\supset(Z\supset W))\wedge(W\supset(Z\supset W)))\supset(Z\supset W)}\text{i.o.} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[Z\supset\forall Y.Y] \qquad [Z]}{\forall Y.Y}}{W}}{Z\supset W}}{(Z\supset\forall Y.Y)\supset(Z\supset W)} \qquad \dfrac{\dfrac{[W]}{Z\supset W}}{W\supset(Z\supset W)}}{((Z\supset\forall Y.Y)\supset(Z\supset W))\wedge(W\supset(Z\supset W))}}{Z\supset W}$$

where the double line of instantiation overflow hides the following portion of derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\forall X.(((Z \supset \forall Y.Y) \supset X) \wedge (W \supset X)) \supset X}{(((Z \supset \forall Y.Y) \supset W) \wedge (W \supset W)) \supset W} \qquad \dfrac{\mathcal{P}_1 \qquad \mathcal{P}_2}{((Z \supset \forall Y.Y) \supset W) \wedge (W \supset W)}}{W}}{Z \supset W}}{(((Z \supset \forall Y.Y) \supset (Z \supset W)) \wedge (W \supset (Z \supset W))) \supset (Z \supset W)}$$

where $\mathcal{P}_1$ and $\mathcal{P}_2$ are respectively the derivations

$$\dfrac{\dfrac{\dfrac{\dfrac{[((Z \supset \forall Y.Y) \supset (Z \supset W)) \wedge (W \supset (Z \supset W))]}{(Z \supset \forall Y.Y) \supset (Z \supset W)} \qquad [Z \supset \forall Y.Y]}{Z \supset W} \qquad [Z]}{W}}{(Z \supset \forall Y.Y) \supset W}$$

and

$$\dfrac{\dfrac{\dfrac{\dfrac{[((Z \supset \forall Y.Y) \supset (Z \supset W)) \wedge (W \supset (Z \supset W))]}{W \supset (Z \supset W)} \qquad [W]}{Z \supset W} \qquad [Z]}{W}}{W \supset W}$$

Via the alternative translation, we obtain the following derivation $\mathcal{D}^\circ$ in $\mathbf{F_{at}}$, considerably simpler than the previous one:

$$\dfrac{\dfrac{\dfrac{\forall X.(((Z \supset \forall Y.Y) \supset X) \wedge (W \supset X)) \supset X}{(((Z \supset \forall Y.Y) \supset W) \wedge (W \supset W)) \supset W} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[Z \supset \forall Y.Y] \qquad [Z]}{\forall Y.Y}}{W}}{Z \supset W} \qquad [Z]}{W}}{(Z \supset \forall Y.Y) \supset W} \qquad \dfrac{\dfrac{\dfrac{[W]}{Z \supset W} \qquad [Z]}{W}}{W \supset W}}{((Z \supset \forall Y.Y) \supset W) \wedge (W \supset W)}}{W}}{Z \supset W}$$

Observe that the original derivation $\mathcal{D}$ has no redexes; $\mathcal{D}^\star$ has a single redex of the form $(((Z \supset \forall Y.Y) \supset (Z \supset W)) \wedge (W \supset (Z \supset W))) \supset (Z \supset W)$, which is administrative (created by the translation $(\cdot)^\star$); from the results of the present paper it follows that $\mathcal{D}^\star$ reduces to $\mathcal{D}^\circ$; the latter derivation has two redexes of the form $Z \supset W$, which are administrative (created by the translation $(\cdot)^\circ$), and descendant of the single redex in $\mathcal{D}^\star$.

References

[1] F. Ferreira. Comments on predicative logic. *Journal of Philosophical Logic*, 35:1–8, 2006.

[2] F. Ferreira and G. Ferreira. Commuting conversions vs. the standard conversions of the “good” connectives. *Studia Logica*, 92:63–84, 2009.

[3] F. Ferreira and G. Ferreira. Atomic polymorphism. *The Journal of Symbolic Logic*, 78(1):260–274, 2013.

[4] G. Ferreira. Eta-conversions of $\mathbf{IPC}$ implemented in atomic $\mathbf{F}$. *Logic Jnl IGPL*, 25(2):115–130, 2017.

[5] G. Ferreira. Rasiowa-Harrop disjunction property. *Studia Logica*, 105(3):649–664, 2017.

[6] J-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*. Cambridge University Press, 1989.

[7] G. Plotkin. Call-by-name, call-by-value and the λ-calculus. *Theoretical Computer Science*, 1:125–159, 1975.

Affiliations:

José Espírito Santo
Centro de Matemática
Universidade do Minho
4710-057 Braga
Portugal
`jes@math.uminho.pt`

Gilda Ferreira
Centro de Matemática, Aplicações Fundamentais e Investigação Operacional
Faculdade de Ciências da Universidade de Lisboa
Campo Grande, Ed. C6
1749-016 Lisboa
Portugal
`gmferreira@fc.ul.pt`