Notre Dame Journal of Formal Logic
Volume 51, Number 4, 2010

A Syntactic Embedding of Predicate Logic into Second-Order Propositional Logic

Morten H. Sørensen and Paweł Urzyczyn

Abstract We give a syntactic translation from first-order intuitionistic predicate logic into second-order intuitionistic propositional logic IPC2. The translation covers the full set of logical connectives $\wedge$, $\vee$, $\to$, $\bot$, $\forall$, and $\exists$, extending our previous work, which studied the significantly simpler case of the universal-implicational fragment of predicate logic. As corollaries of our approach, we obtain simple proofs of nondefinability of $\exists$ from the propositional connectives and nondefinability of $\forall$ from $\exists$ in the second-order intuitionistic propositional logic. We also show that the $\forall$-free fragment of IPC2 is undecidable.

1 Introduction

The standard textbook example of a PSPACE-complete problem is validity (or satisfiability) for "Quantified Boolean Formulas," that is, classical second-order propositional logic. This is in a visible contrast with the ordinary co-NP-complete propositional calculus. But the expressive power of classical propositional logic with or without propositional quantifiers is identical: every formula with quantifiers is equivalent to a propositional one. In other words, one can express exactly the same properties, although at a significantly different cost.

In the case of intuitionistic logic this difference becomes much more dramatic. Propositional intuitionistic logic is PSPACE-complete [15] and adding propositional quantifiers makes it strictly more expressive and undecidable. There are essentially two proofs of the latter fact. One is due to Gabbay and Sobolev [4; 5; 13] (semantical); the other was given by Löb [7] and is based on a translation from first-order logic. The translation applies to the universal-implicational fragment of first-order classical logic with equality. In fact, the restriction to $\forall$ and $\to$ is not essential and Löb's translation can be applied to first-order intuitionistic logic as well. That was briefly remarked in [7] and worked out by Arts and Dekkers [1].

Received December 30, 2008; accepted March 2, 2010; printed September 21, 2010
2010 Mathematics Subject Classification: Primary, 03B20, 03F03
Keywords: propositional quantification, IPC2
 10.1215/00294527-2010-029

Löb's original translation uses an intermediate language with terms representing second-order propositional formulas and with a special predicate I representing provability in second-order propositional logic, which is expressed by a specific set of axioms. A semantic argument (the axioms are satisfied in a certain extension of any first-order model) is used to ensure correctness of the translation. While this idea is certainly ingenious, the proofs in [7; 1] are quite complicated and not very intuitive.

In [14; 20] we gave a simpler, purely syntactic, translation from a subset of the universal-implicational first-order intuitionistic logic in order to obtain a direct undecidability proof of propositional second-order intuitionistic logic (IPC2). The purpose of this paper is to extend that translation to the full first-order intuitionistic logic (with $\exists$, $\wedge$, $\vee$, and $\bot$). Our approach differs from that of [7; 1] also in that we use natural deduction rather than sequent calculus. We believe that using term assignment (in the spirit of the Curry-Howard isomorphism [14]) makes the argument more transparent and easier to grasp.

As a by-product of our main result we show (Corollary 4.8) that the $\forall$-free fragment of IPC2 is undecidable. This ties in with the recent interest in the second-order existential quantification [2; 3; 9; 17; 18; 22]. Moreover, we provide an analysis of normal forms and a systematic proof-search for IPC2; we think that Proposition 2.8 is of independent interest. As an example we give short syntactic proofs of the non-definability of $\exists$ from the propositional connectives and nondefinability of $\forall$ from $\exists$ (Corollaries 3.2 and 3.5).

2 Propositional Second-Order Logic

The language of intuitionistic second-order propositional logic is defined as in [14, Ch. 11]. Formulas are built from the constant $\bot$ and an infinite supply of propositional variables (written $p, q, \ldots$) using the connectives $\vee$, $\wedge$, and $\to$, and the propositional quantifiers $\exists$ and $\forall$. The rules of inference in Figure 1 include a term assignment, where we leave implicit some type information for simplicity.[1] Later we will sometimes use types as superscripts, writing, for example, M^τ if the type of M is not clear from the context.

Thinking in terms of the Curry-Howard isomorphism, we identify a logical judgment $\Gamma \vdash \varphi$ with a type assignment $\Gamma \vdash M : \varphi$. In particular, we often ignore the difference between Γ as a type environment and Γ as a set of formulas. The reduction rules are standard beta-reductions and commuting conversions (permutations). The full list of reduction rules is given in the Appendix.

Normal forms Various strong normalization proofs for second-order systems can be found in the literature, for example, [6; 8; 10; 16; 19]. To our astonishment, none of these proofs applies exactly to our set of reductions, and only a recent paper saved us the extra work of proving the following.

Proposition 2.1 ([21]) *Our system has the strong normalization property.*

It follows that every provable formula is inhabited by a normal form. We can inductively classify all normal forms into three categories:

Introductions: $\lambda x : \tau . N$, $\Lambda p\, N$, $\langle N_1, N_2 \rangle$, $\mathtt{in}_i(N)$, $[\tau, N]$;

Proper eliminators: x, PN, $P\tau$, $P\{i\}$;

Improper eliminators: $\varepsilon_\varphi(P)$, $\mathtt{case}\ P\ \mathtt{of}\ [x]N_1\ \mathtt{or}\ [y]N_2$, $\mathtt{let}\ P\ \mathtt{be}\ [p, x]\ \mathtt{in}\ N$,

$$\Gamma, x:\tau \vdash x:\tau \qquad \frac{\Gamma \vdash M:\bot}{\Gamma \vdash \varepsilon_\tau(M):\tau}$$

$$\frac{\Gamma, x:\sigma \vdash M:\tau}{\Gamma \vdash (\lambda x{:}\sigma\, M):\sigma \to \tau} \qquad \frac{\Gamma \vdash M:\sigma \to \tau \quad \Gamma \vdash N:\sigma}{\Gamma \vdash (MN):\tau}$$

$$\frac{\Gamma \vdash M:\tau_i}{\Gamma \vdash \mathtt{in}_i(M):\tau_1 \vee \tau_2} \qquad \frac{\Gamma \vdash M:\tau \vee \sigma \quad \Gamma, x:\tau \vdash P:\rho \quad \Gamma, y:\sigma \vdash Q:\rho}{\Gamma \vdash (\mathtt{case}\ M\ \mathtt{of}\ [x]P\ \mathtt{or}\ [y]Q):\rho}$$

$$\frac{\Gamma \vdash M:\tau \quad \Gamma \vdash N:\sigma}{\Gamma \vdash \langle M, N\rangle:\tau \wedge \sigma} \qquad \frac{\Gamma \vdash M:\tau_1 \wedge \tau_2}{\Gamma \vdash M\{i\}:\tau_i}$$

$$(p \notin \mathrm{FV}(\Gamma))\ \frac{\Gamma \vdash M:\sigma}{\Gamma \vdash (\Lambda p\, M):\forall p\, \sigma} \qquad \frac{\Gamma \vdash M:\forall p\, \sigma}{\Gamma \vdash (M\tau):\sigma[p:=\tau]}$$

$$\frac{\Gamma \vdash M:\sigma[p:=\tau]}{\Gamma \vdash [\tau, M]:\exists p\sigma} \qquad \frac{\Gamma \vdash M:\exists p\sigma \quad \Gamma, x:\sigma \vdash N:\rho}{\Gamma \vdash (\mathtt{let}\ M\ \mathtt{be}\ [p, x]\ \mathtt{in}\ N):\rho}\ (p \notin \mathrm{FV}(\Gamma, \rho))$$

Figure 1 Rules of IPC2

where P stands for a proper eliminator and N is an arbitrary normal form. It should be clear that every proper eliminator is obtained from a variable (called its *head variable*) by means of a sequence of applications and projections and thus its type must be a "final" part of the type of the head variable. In contrast, types of improper eliminators can be quite arbitrary.

Suffixes and targets In the simply typed lambda-calculus, every type τ can be written as $\tau = \sigma_1 \to \cdots \to \sigma_k \to p$, where p is a type variable, often called the "target" of τ. Any application beginning with a variable of type τ must be of a "suffix" type $\sigma_i \to \cdots \to \sigma_k \to p$, for some i, or just of type p. Another simple observation is that an atomic type is inhabited in an environment Γ only if it is a target of one of the types in Γ.

In the presence of other connectives and quantifiers, this must be properly generalized. For every type τ, we define the set $S(\tau)$ of *suffixes* of τ as the least set such that

1. $\tau \in S(\tau)$;
2. if $\alpha \to \beta \in S(\tau)$, then $\beta \in S(\tau)$;
3. if $\alpha \wedge \beta \in S(\tau)$, then $\alpha, \beta \in S(\tau)$;
4. if $\forall p\, \alpha \in S(\tau)$, then $\alpha[p := \beta] \in S(\tau)$, for all types β.

Clearly, we have the following lemma.

Lemma 2.2 *If $\varphi \in S(\psi)$, then $S(\varphi) \subseteq S(\psi)$.*

The next lemma states a direct characterization of suffixes.

Lemma 2.3

1. $S(\bot) = \{\bot\}$ *and* $S(p) = \{p\}$.
2. $S(\alpha \to \beta) = \{\alpha \to \beta\} \cup S(\beta)$.

3. $S(\alpha \wedge \beta) = \{\alpha \wedge \beta\} \cup S(\alpha) \cup S(\beta)$.
4. $S(\alpha \vee \beta) = \{\alpha \vee \beta\}$.
5. $S(\forall p\, \alpha) = \{\forall p\, \alpha\} \cup \bigcup\{S(\alpha[p := \beta]) \mid \beta \textit{ is a type}\}$.
6. $S(\exists p\, \alpha) = \{\exists p\, \alpha\}$.

Proof In each part, the inclusion from left to right is shown by induction with respect to the definition of S. The opposite direction follows from Lemma 2.2. □

For every τ we also define the set $T(\tau)$ of *targets* of τ. Targets of a type are always *atoms*, that is, propositional variables or $\bot$. The symbol $\mathbb{A}$ below stands for the (infinite) set of all atoms.

1. $T(\bot) = \{\bot\}$ and $T(p) = \{p\}$, for a type variable.
2. $T(\alpha \to \beta) = T(\beta)$.
3. $T(\alpha \diamond \beta) = T(\alpha) \cup T(\beta)$, for $\diamond \in \{\wedge, \vee\}$.
4. $T(\forall p\, \alpha) = \begin{cases} \mathbb{A}, & \text{if } p \in T(\alpha); \\ T(\alpha), & \text{otherwise.} \end{cases}$
5. $T(\exists p\, \alpha) = \begin{cases} \mathbb{A}, & \text{if } T(\alpha) = \mathbb{A}; \\ T(\alpha) - \{p\}, & \text{otherwise.} \end{cases}$

Note that if $T(\tau) \neq \mathbb{A}$ then $T(\tau) \subseteq \mathrm{FV}(\tau) \cup \{\bot\}$; in particular, $T(\tau)$ is finite. The correctness of the above definition of $T(\tau)$ (invariance with respect to alpha-conversion) follows from the next lemma, which, strictly speaking, should itself be part of the definition.

Lemma 2.4

$$T(\alpha[p := \sigma]) = \begin{cases} (T(\alpha) - \{p\}) \cup T(\sigma), & \textit{if } p \in T(\alpha) \neq \mathbb{A}; \\ T(\alpha), & \textit{otherwise.} \end{cases} \qquad (*)$$

In particular, if q is a target of $\alpha[p := \sigma]$, then either p or q is a target of α.

Proof Induction with respect to α. The nonobvious cases are when α begins with a quantifier. Let $\alpha = \forall q\, \beta$, where we can assume $p \neq q \notin \mathrm{FV}(\sigma)$. From the induction hypothesis we know, in particular, that $T(\beta[p := \sigma]) = \mathbb{A}$ if and only if either $T(\beta) = \mathbb{A}$ or $T(\sigma) = \mathbb{A}$ (with $p \in T(\beta)$). In these cases we have $\mathbb{A}$ at both sides of the equation (*).

The same happens when $q \in T(\beta)$, so we are left with two cases to consider. One is when $p, q \notin T(\beta) \neq \mathbb{A}$, and then we have $T(\beta)$ on both sides of (*). The other case is when $T(\sigma) \neq \mathbb{A}$, and $p \in T(\beta)$, but $q \notin T(\beta)$; in particular, $T(\beta) \neq \mathbb{A}$. We know that $q \notin \mathrm{FV}(\sigma)$, and this implies $q \notin T(\sigma)$ (as otherwise $T(\sigma) = \mathbb{A}$). Therefore, $q \notin T(\beta[p := \sigma])$, and, by definition, $T(\alpha[p := \sigma]) = T(\beta[p := \sigma])$ and $T(\alpha) = T(\beta)$. Hence the equation (*) follows immediately from the induction hypothesis.

Now let $\alpha = \exists q\, \beta$. As in the previous case, we have $\mathbb{A}$ on both sides of (*) when either $T(\beta) = \mathbb{A}$ or $T(\sigma) = \mathbb{A}$, with $p \in T(\beta)$. So assume that $T(\beta[p := \sigma]), T(\beta) \neq \mathbb{A}$, whence $T(\alpha[p := \sigma]) = T(\beta[p := \sigma]) - \{q\}$ and $T(\alpha) = T(\beta) - \{q\}$ by definition. If $p \notin T(\beta)$, then $T(\beta[p := \sigma]) = T(\beta)$ and (*) follows easily. If $p \in T(\beta)$, then $T(\sigma) \neq \mathbb{A}$, and it remains to verify the equation

$$((T(\beta) - \{p\}) \cup T(\sigma)) - \{q\} = ((T(\beta) - \{q\}) - \{p\}) \cup T(\sigma),$$

using the fact that $q \notin \mathrm{FV}(\sigma) \supseteq T(\sigma)$. □

Lemma 2.5 *If $\alpha \in S(\psi)$, then $T(\alpha) \subseteq T(\psi)$. In particular, $S(\psi) \cap \mathbb{A} \subseteq T(\psi)$.*

Proof We say that ψ' is an *instance* of ψ when $\psi' = \psi[\vec{p} := \vec{\beta}]$ for some variables $\vec{p} \notin T(\psi)$ and some types $\vec{\beta}$. Note that by Lemma 2.4 we then have $T(\psi') = T(\psi)$.

By induction with respect to ψ we prove that if $\alpha \in S(\psi')$ for some instance ψ' of ψ then $T(\alpha) \subseteq T(\psi)$. Most cases are immediate; we consider the two quantifiers.

Let $\psi = \forall q\, \sigma$. First note that an instance ψ' of ψ must be of the form $\psi' = \forall q\, \sigma'$, where σ' is an instance of σ. This is because $\vec{p} \notin T(\psi)$ implies $\vec{p} \notin T(\sigma)$. Let $\alpha \in S(\psi')$. If $\alpha = \psi'$, then $T(\alpha) = T(\psi)$ as already observed, so we can assume $\alpha \in S(\sigma'[q := \beta])$. If $q \notin T(\sigma)$, then $\sigma'[q := \beta]$ is an instance of σ and by the induction hypothesis we have $T(\alpha) \subseteq T(\sigma) = T(\psi)$. But if $q \in T(\sigma)$, then $T(\psi) = \mathbb{A}$, so the conclusion is immediate.

If $\psi = \exists q\, \sigma$ and $\alpha \in S(\psi')$, then $\alpha = \psi'$ and again we have $T(\alpha) = T(\psi)$. □

If Γ is an environment, then $T(\Gamma)$ is the union of $T(\sigma)$ for all σ declared in Γ.

Lemma 2.6

1. *If $\Gamma, x : \tau \vdash P : \sigma$, and P is a proper eliminator beginning with x, then $\sigma \in S(\tau)$.*
2. *If $\Gamma \vdash a$, where a is an atom, then either $a \in T(\Gamma)$, or $\bot \in T(\Gamma)$ and $\Gamma \vdash \bot$.*

Proof (1) Easy induction with respect to P.

(2) Induction with respect to the size of a normal proof M of a. Since a is an atom, the term M cannot be an introduction, and if it is a proper eliminator then part (1) applies together with Lemma 2.5. By a similar argument, if $M = \varepsilon(P)$ then $\Gamma \vdash P : \bot$ and $\bot \in T(\Gamma)$. Now let $M = \texttt{case}\ P^{\alpha\vee\beta}\ \texttt{of}\ [x]N\ \texttt{or}\ [y]R$. By the induction hypothesis for N (respectively, R) we have either a or $\bot$ in $T(\Gamma) \cup T(\alpha)$ (respectively, $T(\Gamma) \cup T(\beta)$). But $T(\alpha), T(\beta) \subseteq T(\Gamma)$, by Lemma 2.5, because $\alpha \vee \beta \in S(\Gamma)$. Thus either a or $\bot$ is in $T(\Gamma)$. If it is a, then we are done. If, however, $a \notin T(\Gamma)$, then the induction hypothesis yields $\Gamma, \alpha \vdash \bot$ as well as $\Gamma, \beta \vdash \bot$, whence $\Gamma \vdash \bot$. The case $M = \texttt{let}\ P\ \texttt{be}\ [p, x]\ \texttt{in}\ N$ is treated similarly. □

It follows from the above lemma that if $\bot \notin T(\Gamma)$, then $\Gamma \nvdash \bot$; that is, Γ is consistent. Lemmas 2.5 and 2.6 together imply that if $\Gamma \vdash P : \sigma$, with proper P, then $T(\sigma) \subseteq T(\Gamma)$, that is, that proper eliminators do not produce new targets.

Lemma 2.7 *If $q, \bot \notin T(\Gamma)$ and $\Gamma, \varphi \to q \vdash q$, then $\Gamma, \varphi \to q \vdash \varphi$.*

Proof Consider the shortest normal proof of q. It must be an eliminator, and if it is proper, then by Lemma 2.6(1) it must be of the form yM, where y is the assumption of type $\varphi \to q$. Then of course M proves φ.

An improper eliminator beginning with ε is excluded by the consistency of $\Gamma, \varphi \to q$. If the proof is of the form $\texttt{case}\ P^{\alpha\vee\beta}\ \texttt{of}\ [x]Q\ \texttt{or}\ [y]R$ then we have $\Gamma, \varphi \to q, \alpha \vdash Q : q$ and $\Gamma, \varphi \to q, \beta \vdash R : q$. By Lemmas 2.5 and 2.6, types α and β do not introduce new targets, so we still have $q, \bot \notin T(\Gamma, \alpha)$ and $q, \bot \notin T(\Gamma, \beta)$, and we can apply the induction hypothesis to R and Q. Therefore, $\Gamma, \varphi \to q, \alpha \vdash \varphi$ and $\Gamma, \varphi \to q, \beta \vdash \varphi$. Since $\Gamma, \varphi \to q \vdash \alpha \vee \beta$, we conclude that $\Gamma, \varphi \to q \vdash \varphi$.

If the proof is of the form `let` $P^{\exists p\,\tau}$ `be` $[p,\ x]$ `in` N, then we apply the induction hypothesis to the proof $\Gamma, \varphi \to q, \tau \vdash N : q$. We obtain $\Gamma, \varphi \to q, \tau \vdash \varphi$ and thus also $\Gamma, \varphi \to q \vdash \varphi$, because $\Gamma, \varphi \to q \vdash \exists p\,\tau$. □

Indirect targets and splits A suffix of a formula is *weak* when it is of the form $\alpha \vee \beta$ or $\exists p\,\alpha$. A target of a weak suffix of σ is called an *indirect target* of σ. The set of all indirect targets of σ is denoted by $I(\sigma)$. It follows from Lemma 2.5 that $I(\sigma) \subseteq T(\sigma)$; that is, indirect targets are indeed targets. Of course, $I(\Gamma)$ stands for the union of all $I(\sigma)$ where $\sigma \in \Gamma$.

If $\Gamma \vdash \exists\vec{p}\,(\sigma_1 \vee \cdots \vee \sigma_n)$, where $\vec{p}$ are fresh variables, $\Gamma, \sigma_i \nvdash \bot$, and $T(\sigma_i) \subseteq I(\Gamma) \cup \vec{p}$, for each σ_i, then we say that the formula $\exists\vec{p}\,(\sigma_1 \vee \cdots \vee \sigma_n)$ is a *split of* Γ. Formulas σ_i are called *components* of the split. For every consistent Γ there is a *trivial split* of the form $\exists p\, p$.

The Wajsberg/Ben-Yelles algorithm [14] for the simply typed lambda-calculus uses the fact that a normal inhabitant must either be an abstraction (an introduction) or an application (a proper eliminator). We have a weaker form of this property; namely, a type is inhabited by an introduction or a proper eliminator in every component of a certain split. More precisely, we have the following.

Proposition 2.8 *Assume that* $\Gamma \nvdash \bot$*, and let* $\Gamma \vdash \zeta$*, where* ζ *is any formula. There exists a split* $\exists\vec{p}\,(\sigma_1 \vee \cdots \vee \sigma_n)$ *of* Γ *such that, for every* i*, we have* $\Gamma, \sigma_i \vdash N_i : \zeta$ *with* N_i *being either an introduction or a proper eliminator.*

Proof We proceed by induction with respect to the size of a normal inhabitant M of ζ. If M is an introduction or a proper eliminator, then the thesis holds with a trivial split. Since Γ is consistent, M is not of the form $\varepsilon(P)$.

Assume that $M =$ `case` P `of` $[x]Q$ `or` $[y]R$, where P is a proper eliminator of type $\alpha \vee \beta$. Then we have $\Gamma \vdash \alpha \vee \beta$ and $\Gamma, x{:}\alpha \vdash Q : \zeta$ and $\Gamma, y{:}\beta \vdash R : \zeta$.

If $\Gamma, \alpha \vdash \bot$ then we actually have $\Gamma \vdash \beta$; in particular, $\Gamma, \beta \nvdash \bot$. By the induction hypothesis, there is a split $\Gamma, \beta \vdash \exists\vec{p}\,(\rho_1 \vee \cdots \vee \rho_l)$ such that $\Gamma, \beta \wedge \rho_i \vdash Q_i : \zeta$, for all i and no Q_i is improper. Then the formula $\exists\vec{p}\,((\beta \wedge \rho_1) \vee \cdots \vee (\beta \wedge \rho_l))$ is the required split of Γ (note that $T(\beta) \subseteq I(\Gamma)$, because P is proper, and its type is a weak suffix).

The case $\Gamma, \beta \vdash \bot$ is analogous, so let us suppose that neither $\Gamma, \alpha \vdash \bot$ nor $\Gamma, \beta \vdash \bot$. Then the induction hypothesis yields two splits $\Gamma, \alpha \vdash \exists\vec{r}\,(\tau_1 \vee \cdots \vee \tau_k)$ and $\Gamma, \beta \vdash \exists\vec{q}\,(\rho_1 \vee \cdots \vee \rho_l)$ such that $\Gamma, \alpha \wedge \tau_i \vdash \zeta$ and $\Gamma, \beta \wedge \rho_j \vdash \zeta$ hold by either introductions or proper eliminators. Then we can use the split $\exists\vec{r}\vec{q}\,((\alpha \wedge \tau_1) \vee \cdots \vee (\alpha \wedge \tau_k) \vee (\beta \wedge \rho_1) \vee \cdots \vee (\beta \wedge \rho_l))$.

Now let $M =$ `let` P `be` $[q,\ x]$ `in` N, where $\Gamma \vdash P : \exists q.\alpha$. From the induction hypothesis we have a split $\exists\vec{p}\,(\sigma_1 \vee \cdots \vee \sigma_n)$ of Γ, α such that $\Gamma, \alpha \wedge \sigma_i \vdash P_i : \zeta$ with P_i proper eliminators or introductions. We obtain a new split of Γ of the form $\exists q\,\vec{p}\,((\alpha \wedge \sigma_1) \vee \cdots \vee (\alpha \wedge \sigma_n))$. □

3 Intermezzo

Before defining our translation, we play a little intermezzo to demonstrate the use of Proposition 2.8. Corollaries 3.2 and 3.5 are not new, but the proofs we know are semantical [12; 22].

Lemma 3.1 *If* $\vdash \alpha \to \forall p(p \vee \neg p)$*, and* $\forall$ *does not occur in* α*, then* $\vdash \alpha \leftrightarrow \bot$.

Proof Assume the contrary. Then $\alpha \nvdash \bot$, and $T(\alpha) \neq \mathbb{A}$, because α has no occurrence of $\forall$. From $\alpha \vdash \forall p(p \vee \neg p)$ it follows that $\alpha \vdash p \vee \neg p$ for p not free in α, in particular, for $p \notin T(\alpha)$. There is a split $\alpha \vdash \exists \vec{p}(\sigma_1 \vee \cdots \vee \sigma_n)$ with $\alpha, \sigma_i \vdash P_i : p \vee \neg p$, where all P_i are either introductions or proper eliminators. However, since p is not a target of α (and thus also not a target of σ_i), proper eliminators are excluded, and we actually have either $\alpha, \sigma_i \vdash p$ or $\alpha, \sigma_i \vdash \neg p$ for each i. Since p is not free in the environment we conclude that either $\alpha, \sigma_i \vdash \forall p\, p$ or $\alpha, \sigma_i \vdash \forall p\, \neg p$; in other words, $\alpha, \sigma_i \vdash \bot$, for all i. Therefore, $\alpha \vdash \bot$. □

Corollary 3.2 *The universal quantifier is not definable from the other connectives in the intuitionistic second-order propositional logic: there is no formula α without $\forall$ such that $\vdash \alpha \leftrightarrow \forall p(p \vee \neg p)$.*

Proof Immediate from Lemma 3.1, as $\forall p(p \vee \neg p) \nvdash \bot$. □

Remark 3.3 Let A stand for the so-called Pitt's quantifier [11; 12]. It follows immediately from Lemma 3.1 that $\mathsf{A}p(p \vee \neg p)$ is just $\bot$. Note that the result of [11] is often misunderstood. Pitt's construction shows that a *model* of second-order logic can be built over the propositional language. But the class of formulas satisfied in this specific model is a proper extension of IPC2. Therefore, Pitt's quantifier cannot be taken as a *definition* of $\forall$ (even if we restrict attention to the fragment with open instantiation.)

Lemma 3.4 *If $\Gamma \vdash \exists p\, \beta(p)$ and Γ contains no quantifiers, then $\Gamma \vdash \beta(\sigma_1) \vee \cdots \vee \beta(\sigma_n)$, for some $\sigma_1, \ldots, \sigma_n$.*

Proof Induction with respect to the length of a normal proof. The only interesting case is $\Gamma \vdash \texttt{case}\ P^{\gamma \vee \delta}\ \texttt{of}\ [x]Q\ \texttt{or}\ [y]R : \exists p\, \beta(p)$ where we apply induction to Q and R obtaining $\Gamma, \gamma \vdash \beta(\sigma_1) \vee \cdots \vee \beta(\sigma_n)$ and $\Gamma, \delta \vdash \beta(\sigma_{n+1}) \vee \cdots \vee \beta(\sigma_m)$. Clearly, $\Gamma \vdash \beta(\sigma_1) \vee \cdots \vee \beta(\sigma_m)$. Other cases are left to the reader. □

Corollary 3.5 *The existential quantifier is not definable from the propositional connectives in the intuitionistic second-order propositional logic: there is no propositional formula α such that $\vdash \alpha \leftrightarrow \exists q((p \to (\neg q \vee q)) \to p)$.*

Proof Write $\beta(p, q)$ for $(p \to (\neg q \vee q)) \to p$, and assume that $\vdash \alpha \leftrightarrow \exists q\, \beta(p, q)$. By Lemma 3.4, we have $\alpha \vdash \beta(p, \sigma_1) \vee \cdots \vee \beta(p, \sigma_n)$, for some $\sigma_1, \ldots, \sigma_n$. It follows that we also have $\exists q\, \beta(p, q) \vdash \beta(p, \sigma_1) \vee \cdots \vee \beta(p, \sigma_n)$, and even simpler, $\beta(p, q) \vdash \beta(p, \sigma_1) \vee \cdots \vee \beta(p, \sigma_n)$, where q does not occur in σ_i. Since no suffix of $\beta(p, q)$ is a disjunction, we easily observe that a normal proof of $\beta(p, \sigma_1) \vee \cdots \vee \beta(p, \sigma_n)$ must be an introduction. Thus one of the components is provable; that is, we have $\beta(p, q) \vdash \beta(p, \sigma)$, for some σ, not containing q. Therefore,

$$(p \to (\neg q \vee q)) \to p,\ p \to (\neg \sigma \vee \sigma) \vdash p.$$

By induction with respect to the length of a normal proof, we show that this cannot happen. Of course, a normal proof of p cannot be an introduction. An improper eliminator using ε is excluded because $\bot$ is not a suffix. A `case` eliminator requires a shorter proof of p (necessary to reach $\neg \sigma \vee \sigma$) and is excluded by induction. Consider the case of a proper eliminator. Then

$$(p \to (\neg q \vee q)) \to p,\ p \to (\neg \sigma \vee \sigma),\ p \vdash \neg q \vee q,$$

and, therefore, also $\neg\sigma \vee \sigma,\ p \vdash \neg q \vee q$.

The environment $\neg\sigma \vee \sigma,\ p$ is consistent (otherwise, $p \vdash \neg(\neg\sigma \vee \sigma)$, whence $p \vdash \bot$) so we can apply Proposition 2.8. Consider an appropriate split $\neg\sigma \vee \sigma,\ p \vdash \exists\vec{q}(\sigma_1 \vee \cdots \vee \sigma_n)$. The proofs $\neg\sigma \vee \sigma,\ p,\ \sigma_i \vdash \neg q \vee q$ cannot be proper eliminators (q is not a target) so for each i we either have $\neg\sigma \vee \sigma,\ p,\ \sigma_i \vdash \neg q$ or $\neg\sigma \vee \sigma,\ p,\ \sigma_i \vdash q$. If the former case holds for all i, then we actually have $\neg\sigma \vee \sigma,\ p \vdash \neg q$. But the environment $\neg\sigma \vee \sigma,\ p,\ q$ is consistent, by an argument similar to the one above, so we must have $\neg\sigma \vee \sigma,\ p,\ \sigma_i \vdash q$ at least once. This, however, contradicts Lemma 2.6(2). □

4 The Translation

Our source language is intuitionistic first-order logic over a signature consisting of a finite number of binary predicate symbols $\mathtt{P}, \mathtt{Q}, \ldots$. The restriction to binary predicates is not essential and our coding can easily be adopted to arbitrary arities.

The target language is IPC2 of Section 2. As in [14], we assume that all individual variables (written $a, b, \ldots$) can be used as propositional variables (type variables) in the target language. The plan is to systematically replace any atom $\mathtt{P}(a,b)$ in a given first-order formula φ by a certain type $\overline{\mathtt{P}(a,b)}$, to obtain a type $\overline{\varphi}$ such that $\vdash \varphi$ is equivalent to $\vdash \overline{\varphi}$. The difficulty is to ensure that $\overline{\varphi}$ is not provable in an "ad hoc" way. A most naïve attempt could be, for instance, to take $\overline{\mathtt{P}(a,b)} = a \to b \to p$, for some p. The obvious confusion of $\overline{\mathtt{P}(a,b)}$ being equivalent to $\overline{\mathtt{P}(b,a)}$ can be easily fixed, but here is a serious problem: the formula $\exists b \forall a\, \mathtt{P}(a,b)$ is provable, because the variable b can be instantiated by p. Our principal concern is to avoid such ad hoc instantiations.

The solution might be to relativize all quantifiers in $\overline{\varphi}$ using a condition $\mathcal{U}$ such that $\mathcal{U}(A)$ is inhabited only when A is an individual variable (i.e., $\mathcal{U}$ defines the universe of individuals). We cannot do exactly this, but we can ensure a slightly weaker property: a type A satisfying $\mathcal{U}(A)$ must behave (to a sufficient level) as an individual variable (Lemma 4.3).

To define the translation we need some additional type variables:

1. Three variables: $\mathtt{p}$, $\mathtt{p}_1$, and $\mathtt{p}_2$, for each binary relation symbol $\mathtt{P}$;
2. And four more variables: $\bullet$, $\circ$, $\triangledown$, and $\star$.

For an arbitrary type A we write $A^\bullet$ for $A \to \bullet$. If $\mathtt{P}$ is a binary relation symbol, and A, B are arbitrary types, then we define[2]

$$\mathtt{p}_{AB} = (A^\bullet \to \mathtt{p}_1) \to (B^\bullet \to \mathtt{p}_2) \to \mathtt{p};$$
$$\mathtt{p}(A,B) = \mathtt{p}_{AB} \vee \star.$$

For every type A, let $\mathcal{U}(A)$ be the conjunction of all types of the form

$$(A^\bullet \to \mathtt{p}_i) \to \circ \quad \text{and} \quad A^\bullet \to \triangledown,$$

where $i = 1, 2$. As mentioned, the intended meaning of $\mathcal{U}$ is to define the universe of individuals. First-order quantifiers are encoded as second-order quantifiers relativized to $\mathcal{U}$.

The idea of the above definition is to "hide" the type A inside $\mathcal{U}(A)$ deep enough and to consider environments where $\mathcal{U}(a)$ is assumed for every individual variable a. Then an "ad hoc" proof of $\mathcal{U}(A)$ can only be obtained for a type A which is "represented" (see below) by an individual variable.

For every first-order formula φ, we define a second-order propositional formula $\overline{\varphi}$ as follows:

1. $\overline{\mathtt{P}(a,b)} = \mathtt{p}(a,b)$; that is, $\overline{\mathtt{P}(a,b)} = ((a^\bullet \to \mathtt{p}_1) \to (b^\bullet \to \mathtt{p}_2) \to \mathtt{p}) \vee \star$;
2. $\overline{\bot} = \star$;
3. $\overline{\vartheta \diamond \psi} = \overline{\vartheta} \diamond \overline{\psi}$, where $\diamond \in \{\to, \wedge, \vee\}$;
4. $\overline{\forall a\, \psi} = \forall a(\mathcal{U}(a) \to \overline{\psi})$;
5. $\overline{\exists a\, \psi} = \exists a(\mathcal{U}(a) \wedge \overline{\psi})$.

An individual variable a *represents* a type A in an environment Γ if and only if the conditions

$$\Gamma, A^\bullet \vdash a^\bullet,$$
$$\Gamma, A^\bullet \to \mathtt{p}_i \vdash a^\bullet \to \mathtt{p}_i,$$

hold for every relation symbol $\mathtt{P}$ and every $i \in \{1, 2\}$. Note that a variable represents itself.

Lemma 4.1 *Let us fix two atoms of the form* $\mathtt{p}_i$, $\mathtt{q}_j$. *Assume that no individual variable nor any of the symbols* $\bullet$, $\bot$, $\mathtt{p}_i$, $\mathtt{q}_j$ *is in* $T(\Gamma)$. *If*

$$\Gamma, A^\bullet \vdash a^\bullet,$$
$$\Gamma, A^\bullet \to \mathtt{p}_i \vdash b^\bullet \to \mathtt{q}_j,$$

then $a = b$, $\mathtt{p} = \mathtt{q}$, *and* $i = j$.

Proof From $\Gamma, A^\bullet \vdash a^\bullet$ we obtain $\Gamma, a^\bullet \to \mathtt{p}_i \vdash A^\bullet \to \mathtt{p}_i$. Therefore, $\Gamma, a^\bullet \to \mathtt{p}_i \vdash b^\bullet \to \mathtt{q}_j$ and thus $\Gamma, x : a^\bullet \to \mathtt{p}_i,\ y : b^\bullet \vdash N : \mathtt{q}_j$, for some normal form N. Since $\mathtt{q}_j, \bot \notin T(\Gamma)$, we must have $\mathtt{q}_j = \mathtt{p}_i$ because of Lemma 2.6(2). Similarly, $\mathtt{p}_i \notin T(\Gamma, b^\bullet)$, so by Lemma 2.7 we have $\Gamma, a^\bullet \to \mathtt{p}_i, b^\bullet \vdash a^\bullet$, that is, $\Gamma, a^\bullet \to \mathtt{p}_i, b \to \bullet, a \vdash \bullet$. Applying again Lemma 2.7, we conclude that $\Gamma, a^\bullet \to \mathtt{p}_i, b \to \bullet, a \vdash b$. The only individual variable in $T(\Gamma, a^\bullet \to \mathtt{p}_i, b \to \bullet, a)$ is a, so it must be the case that $a = b$. □

Lemma 4.2 *Assume that no individual variable and no variable of the form* $\mathtt{p}_i$ *nor any of the symbols* $\bullet$, $\bot$ *belongs to* $T(\Gamma)$. *If a type* A *is represented in* Γ *by variables* a *and* b *then* $a = b$.

Proof Immediate from Lemma 4.1. □

Note that if $\Gamma \subseteq \Gamma'$ and both the environments satisfy the assumptions of Lemma 4.2, then the variable representing a type A in Γ and Γ' is the same.

Lemma 4.3 *Assume that* Γ *is an environment such that*

1. *individual variables, variables of the form* $\mathtt{p}_i$, *types* $\bot$, *and* $\bullet$ *do not belong to* $T(\Gamma)$,
2. *if* $\circ \in T(\psi)$ *or* $\triangledown \in T(\psi)$, *for some* $\psi \in \Gamma$, *then* $\psi = \mathcal{U}(a)$, *where* a *is an individual variable.*

Suppose that $\Gamma \vdash \mathcal{U}(A)$, *for some type* A. *Then there is a unique individual variable* a *representing* A *in* Γ. *In addition,* Γ *must contain the assumption* $\mathcal{U}(a)$.

Proof Since $\Gamma \vdash \mathcal{U}(A)$, we have $\Gamma \vdash A^\bullet \to \triangledown$; that is, $\Gamma, A^\bullet \vdash \triangledown$. By Proposition 2.8, there is a split $\exists \vec{p}\,(\sigma_1 \vee \cdots \vee \sigma_n)$ of $\Gamma, A^\bullet$ such that $\Gamma, A^\bullet, \sigma_k \vdash P_k : \triangledown$ holds for every k with some proper eliminator P_k. But all targets of σ_k are in $T(\Gamma, A^\bullet) \cup \vec{p}$ and, therefore, the only way in which $\triangledown$ can be a target in $\Gamma, A^\bullet, \sigma_k$ is because some $\mathcal{U}(a)$ is in Γ. Since P_k is proper, we must have $\Gamma, A^\bullet, \sigma_k \vdash a^\bullet$ (Lemma 2.6(1)).

On the other hand, it follows from $\Gamma \vdash \mathcal{U}(A)$ that $\Gamma \vdash (A^\bullet \to \mathrm{p}_i) \to \circ$; that is, $\Gamma, A^\bullet \to \mathrm{p}_i \vdash \circ$. Again, we have a split $\exists \vec{q}\,(\tau_1 \vee \cdots \vee \tau_n)$ of $\Gamma, A^\bullet \to \mathrm{p}_i$ satisfying $\Gamma, A^\bullet \to \mathrm{p}_i, \tau_\ell \vdash P^\ell : \circ$ with proper P^ℓ. The variable $\circ$ may occur in Γ only as target of some $\mathcal{U}(b)$, and we get $\Gamma, A^\bullet \to \mathrm{p}_i, \tau_\ell \vdash b^\bullet \to \mathrm{q}_j$.

For any k and ℓ, the environment $\Gamma, \tau_\ell, \sigma_k$ satisfies the assumptions of Lemma 4.1. This is because, by the definition of split, all targets of τ_ℓ are indirect targets of $\Gamma, A^\bullet \to \mathrm{p}_i$, or are in $\vec{p}$. Since $\mathrm{p}_i \notin T(\Gamma) \cup \vec{p}$, we have that p_i is not a target of τ_ℓ. For a similar reason, $\bullet$ is not a target in $\Gamma, \tau_\ell, \sigma_k$.

From Lemma 4.1 we have that $\mathrm{p}_i = \mathrm{q}_j$ and $a = b$ (in particular, one a is good for every k), and we actually get $\Gamma, A^\bullet \to \mathrm{p}_i, \tau_\ell \vdash a^\bullet \to \mathrm{p}_i$, for all $\ell = 1, \ldots, n$. Since τ_ℓ are components of a split, we conclude that $\Gamma, A^\bullet \to \mathrm{p}_i \vdash a^\bullet \to \mathrm{p}_i$, and, similarly, $\Gamma, A^\bullet \vdash a^\bullet$. It follows that a represents A. Uniqueness is a consequence of Lemma 4.2. □

We say that an environment Γ is *simple* when Γ consists of

1. formulas of the form $\mathcal{U}(a)$, where a is an individual variable;
2. formulas of the form $\overline{\varphi}[\vec{a} := \vec{A}]$ (written $\overline{\varphi}(\vec{A})$ for simplicity), where $\vec{a}$ are individual variables and $\vec{A}$ are arbitrary types called *ad hoc types* of Γ.

Note that the parsing of a type of the form $\overline{\varphi}(\vec{A})$ is unique in the following sense: if we have $\overline{\varphi}(\vec{A}) = \overline{\psi}(\vec{B})$ and no free individual variable occurs twice in φ or ψ then $\vec{B}$ is a permutation of $\vec{A}$, and φ is identical to ψ modulo a renaming of variables. Note also that, no matter what $\vec{A}$ is, the targets of $\overline{\varphi}(\vec{A})$ are only $\star$, and variables of the form q, where Q is a relation symbol. Therefore, only $\star, \mathrm{q}, \circ, \triangledown$ may be targets in a simple environment. It follows that simple environments satisfy the assumptions of Lemma 4.3.

Notice also that a suffix (type of a proper eliminator) in a simple environment is either of the form $\overline{\varphi}(\vec{A})$ or of the form $\mathcal{U}(B) \to \overline{\varphi}(\vec{A}, B)$ or is a suffix of some $\mathcal{U}(a)$. In particular, a variable of the form p cannot be a suffix.

An environment Γ' is a *variant* of Γ when every formula in Γ' is either a member of Γ or a conjunction of formulas in Γ.

Lemma 4.4 *Let $\Delta = \Gamma \cup \Sigma$, where*

1. *Γ is a variant of a simple environment;*
2. *Σ consists exclusively of types of the form q_{CD}, where C and D are represented in Δ by individual variables.*

Assume that $\Delta \vdash \mathrm{p}_{AB}$, where A and B are represented in Δ by individual variables. Then there is $\mathrm{p}_{CD} \in \Sigma$ such that A and C are represented in Δ by the same individual variable, and similarly for B and D.

Proof We have $\Delta, A^\bullet \to \mathrm{p}_1, B^\bullet \to \mathrm{p}_2 \vdash M : \mathrm{p}$, for some normal proof M, and we proceed by induction with respect to the size of M. The term M must be an eliminator, and we have the following cases.

Case 1 M is a proper eliminator. Since $\mathtt{p}$ may occur as a suffix only in Σ, we have

$$\Delta, A^\bullet \to \mathtt{p}_1, B^\bullet \to \mathtt{p}_2 \vdash C^\bullet \to \mathtt{p}_1;$$
$$\Delta, A^\bullet \to \mathtt{p}_1, B^\bullet \to \mathtt{p}_2 \vdash D^\bullet \to \mathtt{p}_2,$$

for some C and D with $\mathtt{p}_{CD} \in \Sigma$. Let a, c be the variables representing A, C in Δ. Then

$$\Delta, A^\bullet \to \mathtt{p}_1, B^\bullet \to \mathtt{p}_2 \vdash c^\bullet \to \mathtt{p}_1 \qquad \text{and} \qquad \Delta, A^\bullet, B^\bullet \to \mathtt{p}_2 \vdash a^\bullet,$$

and, therefore, $a = c$, by Lemma 4.1. A similar argument applies to B and D.

Case 2 $M = \mathtt{case}\ P\ \mathtt{of}\ [x]Q\ \mathtt{or}\ [y]N$, where $P : \tau \vee \sigma$. Then

$$\Gamma, \sigma, \Sigma, A^\bullet \to \mathtt{p}_1, B^\bullet \to \mathtt{p}_2 \vdash N : \mathtt{p}.$$

Here N is a normal proof, shorter than M. Since $\vee$ does not occur in $S(\Sigma)$, the proper eliminator P must begin with a variable declared in Γ. The type $\tau \vee \sigma$ is therefore a suffix of Γ (an instance of a formula), and we can assume that $\sigma = \overline{\psi}(\vec{A})$, for some ψ and $\vec{A}$. (It may happen that P is of type $\mathtt{q}(A, B) = \mathtt{q}_{AB} \vee \star$. In this case we assume $\tau = \mathtt{q}_{AB}$ and $\sigma = \star = \overline{\bot}$.)

Thus the environment $\Gamma, x : \sigma$ is simple and we can apply the induction hypothesis to N. It follows that $\mathtt{p}_{CD} \in \Sigma$, where A and C (and also B and D) are represented by the same variable in Δ, σ. From the uniqueness we conclude that these types are represented by the same variable in Δ.

Case 3 $M = \mathtt{let}\ P\ \mathtt{be}\ [a, x]\ \mathtt{in}\ N$. The head variable of the proper eliminator P must be declared in Γ, because an existential formula is a suffix of its type. Thus P is of type $\exists a\, \overline{\varphi}(a, \vec{A})$, where a is an individual variable, and we have

$$\Gamma, \overline{\varphi}(a, \vec{A}), \Sigma, A^\bullet \to \mathtt{p}_1, B^\bullet \to \mathtt{p}_2 \vdash N : \mathtt{p},$$

where N is shorter than M. Again we apply induction.

Case 4 $M = \varepsilon(P)$ is excluded, because $\bot$ is not a target in the environment $\Gamma, \Sigma, A^\bullet \to \mathtt{p}_1, B^\bullet \to \mathtt{p}_2$. □

For a first-order environment Σ, we define

$$\overline{\Sigma} = \{\overline{\varphi} \mid \varphi \in \Sigma\} \cup \{\mathcal{U}(a) \mid a \in \mathrm{FV}(\Sigma)\}.$$

Clearly, $\overline{\Sigma}$ is a simple environment.

Suppose that Γ is a simple environment such that $\Gamma \vdash \mathcal{U}(A)$, for every ad hoc type A of Γ. By Lemma 4.3, the ad hoc types are represented in Γ by individual variables (and these variables occur free in Γ). Thus, we can define the first-order environment

$$|\Gamma| = \{\varphi(\vec{a}) \mid \overline{\varphi}(\vec{A}) \in \Gamma, \text{ for some } \vec{A}, \text{ and variables } \vec{a} \text{ represent } \vec{A} \text{ in } \Gamma\}.$$

Of course, $|\overline{\Sigma}| = \Sigma$ for first-order Σ. Note also that all free variables of $|\Gamma|$ occur free in Γ.

Let Γ' be a variant of a simple environment Γ such that $\Gamma \vdash \mathcal{U}(A)$ for every ad hoc type A of Γ. We say that a union of the form $\Delta = \Gamma' \cup \Sigma$ is a *good environment* (and we write $\Delta \approx \Gamma \oplus \Sigma$), when every type in Σ is of the form $\mathtt{q}_{AB}$, with

1. $\Delta \vdash \mathcal{U}(A)$ and $\Delta \vdash \mathcal{U}(B)$;
2. $|\Gamma| \vdash \mathrm{Q}(a, b)$, for a, b representing A, B in Δ.

Targets of a good environment are only of the form $\star$, q, $\circ$, $\triangledown$, quite like in a simple environment.

Lemma 4.5 *If $\Delta \approx \Gamma \oplus \Sigma$ is a good environment, and $\Delta \vdash P : \sigma$, for a proper eliminator P, then either $\sigma \in \Delta$ or one of the following cases holds:*

1. $\sigma = \overline{\varphi}(\vec{A})$ *and* $\Delta \vdash \mathcal{U}(A)$, *for each* $A \in \vec{A}$;
2. $\sigma = \mathcal{U}(B) \to \overline{\varphi}(\vec{A}, B)$, *where* $\Delta \vdash \mathcal{U}(A)$, *for each* $A \in \vec{A}$, *and* $P = P'B$, *for some* P';
3. $\sigma = \sigma_1 \wedge \sigma_2$, *where* $\sigma_1, \sigma_2 \in \Gamma$;
4. $\sigma \in S(\mathcal{U}(a))$, *for some individual variable* a;
5. $\sigma \in S(\mathrm{p}_{AB})$, *for some* $\mathrm{p}_{AB} \in \Sigma$.

Proof Induction with respect to the length of P. □

Here is our main lemma.

Lemma 4.6 *If $\Delta \approx \Gamma \oplus \Sigma$ is good and $\Delta \vdash \overline{\varphi}(\vec{A})$, with $\Delta \vdash \mathcal{U}(A)$ for each $A \in \vec{A}$, then $|\Gamma| \vdash \varphi(\vec{a})$, in first-order logic, where $\vec{a}$ represent $\vec{A}$ in Δ.*

Proof We prove a slightly more general statement, consisting of three claims (where M is assumed normal, the variables $\vec{a}$ represent $\vec{A}$, and $\Delta \vdash \mathcal{U}(A)$ for all A in $\vec{A}$):

(a) If $\Delta \vdash M : \overline{\varphi}(\vec{A})$, then $|\Gamma| \vdash \varphi(\vec{a})$.
(b) If $\Delta \vdash M : \mathcal{U}(a) \to \overline{\varphi}(a, \vec{A})$, where a is not free in Δ, then $|\Gamma| \vdash \forall a\, \varphi(a, \vec{a})$.
(c) If $\Delta \vdash M : \mathcal{U}(A) \wedge \overline{\varphi}(A, \vec{A})$, then $|\Gamma| \vdash \varphi(a, \vec{a})$, where a represents A.

We proceed by induction with respect to M by inspecting the various forms M may have. In each case we consider the relevant claims among (a)–(c).

Case 1 M is an abstraction. The relevant subcases are (a) and (b). If M in part (a) is an abstraction of type $\overline{\varphi}(\vec{A})$, then $\overline{\varphi}(\vec{A}) = \overline{\psi}(\vec{A}) \to \overline{\vartheta}(\vec{A})$ and we have $M = \lambda x : \overline{\psi}(\vec{A}).N$, where N is such that $\Delta,\ x : \overline{\psi}(\vec{A}) \vdash N : \overline{\vartheta}(\vec{A})$. The environment $\Delta,\ x : \overline{\psi}(\vec{A})$ is good, because $\Gamma \vdash \mathcal{U}(A)$ holds for each $A \in \vec{A}$. From the induction hypothesis we obtain $|\Gamma|,\ x : \psi(\vec{a}) \vdash \vartheta(\vec{a})$, whence also $|\Gamma| \vdash \psi(\vec{a}) \to \vartheta(\vec{a})$.

If M in (b) is an abstraction $\lambda x{:}\mathcal{U}(a).N$ of type $\mathcal{U}(a) \to \overline{\varphi}(a, \vec{A})$, then $\Gamma,\ x : \mathcal{U}(a) \vdash N : \overline{\varphi}(a, \vec{A})$. We apply the induction hypothesis and obtain $|\Gamma| \vdash \varphi(a, \vec{a})$. Since a is not free in Δ, it is also not free in $|\Gamma|$, and we conclude with $|\Gamma| \vdash \forall a\, \varphi(a, \vec{a})$.

Case 2 M is a polymorphic abstraction. Then we are in part (a) and M is of the form $\Lambda a\, N$ and has type $\overline{\varphi}(\vec{A}) = \forall a\, (\mathcal{U}(a) \to \overline{\psi}(a, \vec{A}))$. Apply part (b) of the induction hypothesis to N.

Case 3 If $M = [A, N]$, then only part (a) is relevant, with $\overline{\varphi}(\vec{A}) = \exists a\, (\mathcal{U}(a) \wedge \overline{\psi}(a, \vec{A}))$ and we have $\Gamma \vdash N : \mathcal{U}(A) \wedge \overline{\psi}(A, \vec{A})$. We apply part (c) of the induction hypothesis to N.

Case 4 M is a pair of the form $\langle N_1, N_2 \rangle$. We consider parts (a) and (c). In part (a) we have $N_1 : \overline{\varphi}(\vec{A})$ and $N_2 : \overline{\psi}(\vec{A})$, and applying induction to N_1 and N_2 we get $|\Gamma| \vdash \varphi(\vec{a})$ and $|\Gamma| \vdash \psi(\vec{a})$. It follows that $|\Gamma| \vdash \varphi(\vec{a}) \wedge \psi(\vec{a})$. In part (c) the pair $\langle N_1, N_2 \rangle$ is of type $\mathcal{U}(A) \wedge \overline{\varphi}(A, \vec{A})$. We apply induction to N_2 and obtain $|\Gamma| \vdash \varphi(a, \vec{a})$.

Case 5 $M = \mathtt{in}_i(N)$. This can only happen in part (a), but we have three subcases. The first subcase is when M is of type $\overline{\varphi}(\vec{A}) \vee \overline{\psi}(\vec{A})$, and it follows easily from the induction hypothesis. The second subcase is when $\Delta \vdash N : \mathtt{p}_{AB}$ and $M = \mathtt{in}_1(N)$ has type $p(A, B) = \mathtt{p}_{AB} \vee \star$. It follows from Lemma 4.4 that there is an assumption $\mathtt{p}_{CD}$ in Σ such that the variables a, b representing A, B in Δ also represent C, D. Therefore, $|\Gamma| \vdash \mathtt{P}(a, b)$. The third subcase is when $\Delta \vdash N : \star$. Since $\star = \overline{\bot}$, the induction hypothesis, part (a), applied to N, implies that $|\Gamma|$ is inconsistent. In particular, $|\Gamma| \vdash \mathtt{P}(a, b)$.

Now we assume that M is a proper eliminator.

Case 6 If M is a variable then the relevant parts are (a) and (c) and the claim is obvious.

Case 7 The case of M being an application is only possible in part (a) and it splits into two subcases. First we assume that $M = PN$, where $\Delta \vdash P : \overline{\psi}(\vec{B}) \to \overline{\varphi}(\vec{A})$. Then $\Delta \vdash \mathcal{U}(B)$ for $B \in \vec{B}$, by Lemma 4.5, and we can apply the induction hypothesis to both P and N. The other subcase is when $M = PBW$, where B is a type. Assume for simplicity that $B \in \vec{A}$, say $\vec{A} = (B, \vec{C})$. Then $\Delta \vdash P : \forall b\, (\mathcal{U}(b) \to \overline{\varphi}(b, \vec{C}))$ and $\Delta \vdash W : \mathcal{U}(B)$. The induction hypothesis (b) applies to Pa, for a fresh a, whence $|\Gamma| \vdash \forall a\, \varphi(a, \vec{c})$ and thus also $|\Gamma| \vdash \varphi(b, \vec{c})$, for b representing B.

Case 8 The case of polymorphic application $M = PB$, where B is a type, is only possible in part (b) and follows immediately from the induction hypothesis (a).

Case 9 If M is a projection, say $M = P\{2\}$, then by Lemma 4.5 we have $\Delta \vdash P : \sigma \wedge \overline{\varphi}(\vec{A})$, for some σ, and either $\overline{\varphi}(\vec{A})$ is in Γ or the induction hypothesis is applicable to P by Lemma 4.5.

There is no other possibility for M to be a proper eliminator, so we now assume that M is improper.

Case 10 If $M = \mathtt{case}\ P\ \mathtt{of}\ [x]Q\ \mathtt{or}\ [y]R$, then (regardless if we are in part (a), (b), or (c)) we have two possibilities. One is that $\Delta \vdash P : \overline{\psi}(\vec{B}) \vee \overline{\vartheta}(\vec{B})$. Then by Lemma 4.5 we can apply induction to P, Q, and R. For instance, in part (a) we then have $|\Gamma| \vdash \psi(\vec{b}) \vee \vartheta(\vec{b})$ and $|\Gamma|, \psi(\vec{b}) \vdash \varphi(\vec{a})$ and $|\Gamma|, \vartheta(\vec{b}) \vdash \varphi(\vec{a})$, for appropriate $\vec{b}$, and therefore also $|\Gamma| \vdash \varphi(\vec{a})$. The argument in parts (b) and (c) is similar. The other possibility is that $\Delta \vdash P : \mathtt{p}_{AB} \vee \star$; that is, P is of type $\mathtt{p}(A, B)$. By part (a) of the induction hypothesis, applied to P, we have $|\Gamma| \vdash \mathtt{P}(a, b)$ for appropriate a, b, whence $\Delta, \mathtt{p}_{AB}$ is good. Thus we can also apply (the appropriate part of) the induction hypothesis to Q, obtaining the desired conclusion.

Case 11 Finally, let $M = \mathtt{let}\ P\ \mathtt{be}\ [b, x]\ \mathtt{in}\ N$ and let us consider part (a). Then M is of type $\overline{\varphi}(\vec{A})$ and $\Delta \vdash P : \exists b\, (\mathcal{U}(b) \wedge \overline{\psi}(b, \vec{B}))$. We also have $\Delta, x{:}\mathcal{U}(b) \wedge \overline{\psi}(b, \vec{B}) \vdash N : \overline{\varphi}(\vec{A})$. We apply induction to P and N and obtain that $|\Gamma| \vdash \exists b\, \psi(b, \vec{b})$ and $|\Gamma|, \psi(b, \vec{b}) \vdash \varphi(\vec{a})$. That is, we have $|\Gamma| \vdash \varphi(\vec{a})$. The reasoning in parts (b) and (c) is similar.

The final remark is that $M \neq \varepsilon(P)$, as $\bot \notin T(\Delta)$. □

Theorem 4.7 *The translation is sound and complete in the following sense: For any first-order* Σ *and* φ, *we have* $\Sigma \vdash \varphi$ *if and only if* $\overline{\Sigma} \vdash \overline{\varphi}$.

Proof The "only if" part goes by a routine induction. (First show that $\overline{\bot} \vdash \overline{\varphi}$, for all φ.) The "if" part is immediate from Lemma 4.6. □

Corollary 4.8 *The ∀-free fragment of intuitionistic second-order propositional logic is undecidable.*

Proof We begin with the ∀-free fragment of classical first-order logic, which is of course undecidable. Via Kolmogorov's translation it reduces to the ∀-free fragment of intuitionistic first-order logic. It remains to observe that our translation does not introduce new universal quantifiers. □

5 Conclusion and Future Work

We have given a purely syntactic translation of first-order intuitionistic logic to second-order intuitionistic propositional logic, thus reproving syntactically the result of [7; 1]. It follows that second-order intuitionistic propositional logic is undecidable and that the same holds for its ∀-free fragment. Note also that for the "only if" part of Theorem 4.7 we only need to instantiate bound variables by variables. That is, undecidability remains true under a strictly predicative regime.

At present, the translation applies to function-free signatures, and the extension to functions remains future work. Another unsettled issue is the exact delineation of the border between decidable and undecidable fragments of ∀-free IPC2. From [18] we know that the ∃, ∧, ¬-fragment is decidable. Decidability with ∀, ∃, ∧, ¬ was also recently announced [17]. The proof of Corollary 4.8 uses ∃, →, ∨, and ∧; it remains open whether all these four connectives are indeed necessary.

The syntactic proof was made possible by an analysis of normal forms in the extended version of system **F**, involving all the logical connectives and quantifiers. This classification appears to be useful on its own, as demonstrated by the simple proofs of nondefinability of ∃ from the propositional connectives, and the nondefinability of ∀ from ∃.

Appendix: Reductions in IPC2

Beta-reductions:

1. $(\lambda x\, M)N \;\Rightarrow\; M[x := N]$;
2. $(\Lambda p\, M)\tau \;\Rightarrow\; M[p := \tau]$;
3. $\langle M_1, M_2\rangle\{i\} \;\Rightarrow\; M_i$;
4. $\texttt{case in}_i(M)\texttt{ of }[x_1]P_1\texttt{ or }[x_2]P_2 \;\Rightarrow\; P_i[x_i := M]$;
5. $\texttt{let }[\tau, M]\texttt{ be }[p, x]\texttt{ in }N \;\Rightarrow\; N[p := \tau][x := M]$.

Commuting conversions for ε*:*

1. $\varepsilon_\psi(\varepsilon_\bot(M)) \;\Rightarrow\; \varepsilon_\psi(M)$;
2. $\varepsilon_{\varphi\to\psi}(M)N \;\Rightarrow\; \varepsilon_\psi(M)$;
3. $\varepsilon_{\forall p.\sigma}(M)\tau \;\Rightarrow\; \varepsilon_{\sigma[p:=\tau]}(M)$;
4. $\varepsilon_{\varphi_1\wedge\varphi_2}(M)\{i\} \;\Rightarrow\; \varepsilon_{\varphi_i}(M)$;
5. $\texttt{case }\varepsilon_{\sigma\vee\tau}(M)\texttt{ of }[u]R^\rho\texttt{ or }[v]S^\rho \;\Rightarrow\; \varepsilon_\rho(M)$;
6. $\texttt{let }\varepsilon_{\exists p.\sigma}(M)\texttt{ be }[p, x]\texttt{ in }N^\rho \;\Rightarrow\; \varepsilon_\rho(M)$;

Commuting conversions for `case`*:*

1. $\varepsilon_\varphi(\texttt{case }M\texttt{ of }[x]P\texttt{ or }[y]Q) \;\Rightarrow\; \texttt{case }M\texttt{ of }[x]\varepsilon_\varphi(P)\texttt{ or }[y]\varepsilon_\varphi(Q)$;
2. $(\texttt{case }M\texttt{ of }[x]P\texttt{ or }[y]Q)N \;\Rightarrow\; \texttt{case }M\texttt{ of }[x]PN\texttt{ or }[y]QN$;

3. ($\mathtt{case}\ M\ \mathtt{of}\ [x]P\ \mathtt{or}\ [y]Q)\tau \Rightarrow \mathtt{case}\ M\ \mathtt{of}\ [x]P\tau\ \mathtt{or}\ [y]Q\tau$;
4. ($\mathtt{case}\ M\ \mathtt{of}\ [x]P\ \mathtt{or}\ [y]Q)\{i\} \Rightarrow \mathtt{case}\ M\ \mathtt{of}\ [x]P\{i\}\ \mathtt{or}\ [y]Q\{i\}$;
5. $\mathtt{case}\ (\mathtt{case}\ M\ \mathtt{of}\ [x]P\ \mathtt{or}\ [y]Q)\ \mathtt{of}\ [u]R\ \mathtt{or}\ [v]S \Rightarrow$
   $\mathtt{case}\ M\ \mathtt{of}\ [x](\mathtt{case}\ P\ \mathtt{of}\ [u]R\ \mathtt{or}\ [v]S)$
   $\mathtt{or}\ [y](\mathtt{case}\ Q\ \mathtt{of}\ [u]R\ \mathtt{or}\ [v]S)$;
6. $\mathtt{let}\ (\mathtt{case}\ M\ \mathtt{of}\ [x]P\ \mathtt{or}\ [y]Q)\ \mathtt{be}\ [p, x]\ \mathtt{in}\ N \Rightarrow$
   $\mathtt{case}\ M\ \mathtt{of}\ [x](\mathtt{let}\ P\ \mathtt{be}\ [p, x]\ \mathtt{in}\ N)$
   $\mathtt{or}\ [y](\mathtt{let}\ Q\ \mathtt{be}\ [p, x]\ \mathtt{in}\ N)$.

Commuting conversions for `let`*:*

1. $\varepsilon_\varphi(\mathtt{let}\ M\ \mathtt{be}\ [p, x]\ \mathtt{in}\ N) \Rightarrow \mathtt{let}\ M\ \mathtt{be}\ [p, x]\ \mathtt{in}\ \varepsilon_\varphi(N)$;
2. $(\mathtt{let}\ M\ \mathtt{be}\ [p, x]\ \mathtt{in}\ N)P \Rightarrow \mathtt{let}\ M\ \mathtt{be}\ [p, x]\ \mathtt{in}\ NP$;
3. $(\mathtt{let}\ M\ \mathtt{be}\ [p, x]\ \mathtt{in}\ N)\tau \Rightarrow \mathtt{let}\ M\ \mathtt{be}\ [p, x]\ \mathtt{in}\ N\tau$;
4. $(\mathtt{let}\ M\ \mathtt{be}\ [p, x]\ \mathtt{in}\ N)\{i\} \Rightarrow \mathtt{let}\ M\ \mathtt{be}\ [p, x]\ \mathtt{in}\ N\{i\}$;
5. $\mathtt{case}\ (\mathtt{let}\ M\ \mathtt{be}\ [p, x]\ \mathtt{in}\ N)\ \mathtt{of}\ [x]P\ \mathtt{or}\ [y]Q \Rightarrow$
   $\mathtt{let}\ M\ \mathtt{be}\ [p, x]\ \mathtt{in}\ \mathtt{case}\ N\ \mathtt{of}\ [x]P\ \mathtt{or}\ [y]Q$;
6. $\mathtt{let}\ (\mathtt{let}\ M\ \mathtt{be}\ [p, x]\ \mathtt{in}\ N)\ \mathtt{be}\ [q, y]\ \mathtt{in}\ P \Rightarrow$
   $\mathtt{let}\ M\ \mathtt{be}\ [p, x]\ \mathtt{in}\ (\mathtt{let}\ N\ \mathtt{be}\ [q, y]\ \mathtt{in}\ P)$.

Notes

1. Strictly speaking, we should write, for example, $[\tau, M]_{\exists p\sigma}$ instead of $[\tau, M]$, etc.

2. This differs from the coding used in [14, Ch. 11], where we had $\mathtt{p}(A, B) = \mathtt{p}_{AB} \to \star$. This coding was appropriate for the restricted class of formulas used there, but does not work in general. Consider, for instance, the unprovable entailment $z : (\mathtt{P}(a, b) \to \mathtt{Q}(c, d)) \to \mathtt{P}(a, b) \vdash \mathtt{P}(a, b)$. The translation of [14] yields the assertion $z : (\mathtt{p}(a, b) \to \mathtt{q}(c, d)) \to \mathtt{p}(a, b) \vdash \mathtt{p}(a, b)$, inhabited by the term $\lambda x^{\mathtt{p}_{ab}}.\, z(\lambda u^{\mathtt{p}(a,b)} \lambda v^{q_{cd}}.\, ux)x$.

References

[1] Arts, T., and W. Dekkers, "Embedding first order predicate logic in second order propositional logic," Technical Report 93-02, Katholieke Universiteit Nijmegen, 1993. 457, 458, 470

[2] Fujita, K., and A. Schubert, "Existential type systems with no types in terms," pp. 112–26 in *Typed Lambda Calculi and Applications*, edited by P.-L. Curien, vol. 5608 of *Lecture Notes in Computer Science*, Springer, Berlin, 2009. Zbl pre05576323. 458

[3] Fujita, K., "Galois embedding from polymorphic types into existential types," pp. 194–208 in *Typed Lambda Calculi and Applications*, edited by P. Urzyczyn, vol. 3461 of *Lecture Notes in Computer Science*, Springer, Berlin, 2005. Zbl 1114.03009. MR 2188768. 458

[4] Gabbay, D. M., "On 2nd order intuitionistic propositional calculus with full comprehension," *Archiv für mathematische Logik und Grundlagenforschung*, vol. 16 (1974), pp. 177–86. Zbl 0289.02016. MR 0360222. 457

[5] Gabbay, D. M., *Semantical Investigations in Heyting's Intuitionistic Logic*, vol. 148 of *Synthese Library*, D. Reidel Publishing Co., Dordrecht, 1981. Zbl 0453.03001. MR 613144. 457

[6] de Groote, P., "On the strong normalisation of intuitionistic natural deduction with permutation-conversions," *Information and Computation*, vol. 178 (2002), pp. 441–64. Zbl 1031.03071. MR 1946174. 458

[7] Löb, M. H., "Embedding first order predicate logic in fragments of intuitionistic logic," *The Journal of Symbolic Logic*, vol. 41 (1976), pp. 705–18. Zbl 0358.02012. MR 0441680. 457, 458, 470

[8] Matthes, R., "Non-strictly positive fixed points for classical natural deduction," *Annals of Pure and Applied Logic*, vol. 133 (2005), pp. 205–30. Zbl 1066.03057. MR 2126159. 458

[9] Nakazawa, K., M. Tatsuta, Y. Kameyama, and H. Nakano, "Undecidability of type-checking in domain-free typed lambda-calculi with existence," pp. 478–92 in *Computer Science Logic. Proceedings of the 22nd International Workshop (CSL 2008, Bertinoro)*, edited by M. Kaminski and S. Martini, vol. 5213 of *Lecture Notes in Computer Science*, Springer, Berlin, 2008. Zbl 1156.03316. 458

[10] Nakazawa, K., and M. Tatsuta, "Strong normalization of classical natural deduction with disjunctions," *Annals of Pure and Applied Logic*, vol. 153 (2008), pp. 21–37. Zbl 1141.03027. MR 2405205. 458

[11] Pitts, A. M., "On an interpretation of second-order quantification in first-order intuitionistic propositional logic," *The Journal of Symbolic Logic*, vol. 57 (1992), pp. 33–52. Zbl 0763.03009. MR 1150924. 463

[12] Połacik, T., "Pitts' quantifiers are not topological quantification," *Notre Dame Journal of Formal Logic*, vol. 39 (1998), pp. 531–44. Zbl 0966.03008. MR 1776225. 462, 463

[13] Sobolev, S. K., "On the intuitionistic propositional calculus with quantifiers," *Matematicheskie Zametki*, vol. 22, (1977), pp. 69–76. Zbl 0365.02013. MR 0457155. 457

[14] Sørensen, M. H., and P. Urzyczyn, *Lectures on the Curry-Howard Isomorphism*, vol. 149 of *Studies in Logic and the Foundations of Mathematics*, Elsevier, Amsterdam, 2006. Zbl 1183.03004. 458, 462, 464, 471

[15] Statman, R., "Intuitionistic propositional logic is polynomial-space complete," *Theoretical Computer Science*, vol. 9 (1979), pp. 67–72. Zbl 0411.03049. MR 535124. 457

[16] Tatsuta, M., "Second-order permutative conversions with Prawitz's strong validity," *Progress in Informatics*, vol. 2 (2005), pp. 41–56. 458

[17] Tatsuta, M., "Second-order system without implication nor disjunction," *Bulletin of Symbolic Logic*, vol. 15 (2009), pp. 262–3. 458, 470

[18] Tatsuta, M., K. Fujita, R. Hasegawa, and H. Nakano, "Inhabitation of existential types is decidable in negation-product fragment," *Proceedings of Second International Workshop on Classical Logic and Computation (CLC2008, Reykjavik)*, 2008. 458, 470

[19] Tatsuta, M., "Simple saturated sets for disjunction and second-order existential quantification," pp. 366–80 in *Typed Lambda Calculi and Applications*, edited by S. Ronchi Della Rocca, vol. 4583 of *Lecture Notes in Computer Science*, Springer, Berlin, 2007. Zbl pre05527391. MR 2391793. 458

[20] Urzyczyn, P., "Inhabitation in typed lambda-calculi (a syntactic approach)," pp. 373–89 in *Typed Lambda Calculi and Applications (Nancy, 1997)*, edited by P. de Groote and J. R. Hindley, vol. 1210 of *Lecture Notes in Computer Science*, Springer, Berlin, 1997. Zbl 1063.03527. MR 1480482. 458

[21] Wojdyga, A., "Short proofs of strong normalization," pp. 613–23 in *Mathematical Foundations of Computer Science 2008*, edited by E. Ochmański and J. Tyszkiewicz, vol. 5162 of *Lecture Notes in Computer Science*, Springer, Berlin, 2008. Zbl 1173.03303. MR 2539405. 458

[22] Zdanowski, K., "On second order intuitionistic propositional logic without a universal quantifier," *The Journal of Symbolic Logic*, vol. 74 (2009), pp. 157–67. Zbl 1163.03010. MR 2499424. 458, 462

Acknowledgments

Partly supported by the Polish Government Grant 3 T11C 002 27. The authors wish to thank Ken-etsu Fujita and Konrad Zdanowski for their comments.

Formalit
Byenden 32
4660 Store Heddinge
DENMARK
mhs@formalit.dk

Institute of Informatics
University of Warsaw
Banacha 2
02-097 Warszawa
POLAND
urzy@mimuw.edu.pl