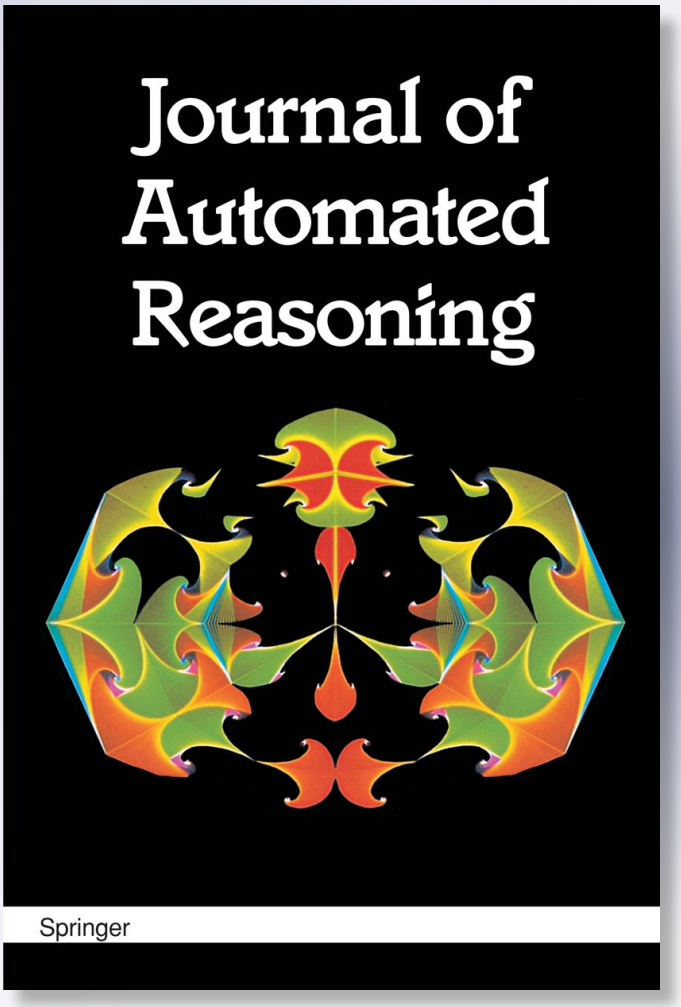
Formalizing Adequacy: A Case Study for Higher-order Abstract Syntax

James Cheney, Michael Norrish & René Vestergaard

Journal of Automated Reasoning

ISSN 0168-7433
Volume 49
Number 2

J Autom Reasoning (2012) 49:209-239
DOI 10.1007/s10817-011-9221-6



Springer

J Autom Reasoning (2012) 49:209–239
DOI 10.1007/s10817-011-9221-6

Formalizing Adequacy: A Case Study for Higher-order Abstract Syntax

James Cheney · Michael Norrish · René Vestergaard

Received: 1 February 2011 / Accepted: 3 February 2011 / Published online: 2 March 2011

Abstract Adequacy is an important criterion for judging whether a formalization is suitable for reasoning about the actual object of study. The issue is particularly subtle in the expansive case of approaches to languages with name-binding. In prior work, adequacy has been formalized only with respect to specific representation techniques. In this article, we give a general formal definition based on model-theoretic *isomorphisms* or *interpretations*. We investigate and formalize an adequate interpretation of untyped lambda-calculus within a higher-order metalanguage in Isabelle/HOL using the Nominal Datatype Package. Formalization elucidates some subtle issues that have been neglected in informal arguments concerning adequacy.

Keywords Adequacy · Isomorphism · Interpretation · Nominal abstract syntax · Higher-order abstract syntax

One can't proceed from the informal to the formal by formal means.
—Alan Perlis

J. Cheney (✉)
Laboratory for Foundations of Computer Science, University of Edinburgh,
Edinburgh, Scotland, UK
e-mail: jcheney@inf.ed.ac.uk

M. Norrish
Canberra Research Lab, NICTA, Canberra, Australia

M. Norrish
Australian National University, Canberra, Australia
e-mail: Michael.Norrish@nicta.com.au

R. Vestergaard
Research Center for Integrated Science, JAIST, Ishikawa, Japan
e-mail: vester@jaist.ac.jp

1 Introduction

The right choice of representation is often the key to success in formal or machine-checked reasoning. A change in representation may make formal, machine-checked proof much easier, but also introduces an additional burden of proof to legitimize reasoning about an object via its representation. The representation must correctly capture the salient properties of the original object of interest. In the context of formal reasoning about languages with name-binding, this correspondence is sometimes called *adequacy*. Reasoning about languages with name-binding, equivalence modulo consistent renaming of bound names (α-equivalence), and capture-avoiding substitution directly is challenging, and there has been extensive exploration of alternative representations. The term "adequacy" was first used to refer to a desired correspondence between an object language and its representation by Harper, Honsell and Plotkin in their seminal work on the higher-order, dependently-typed Logical Framework (LF) [10]. However, as Crary and Harper have pointed out, adequacy is just as important for other formalisms for reasoning about abstract syntax with binding [6].

Adequacy for representations of languages with name-binding is challenging for several reasons. First, a plethora of techniques for representing and reasoning about name-binding and α-equivalence have been introduced, each with different advantages, disadvantages, and caveats. Second, adequacy apparently cannot be proved once and for all, even for a given representational technique. Instead, it seemingly needs to be revisited whenever a new programming language calculus or logic is developed. Third, adequacy proofs are tiresome and are often omitted or conducted on paper using informal approaches to name-binding; thus, they share the well-known disadvantages of paper-and-pencil syntactic proofs. Finally, there remains confusion about the meaning of the term "adequacy" with respect to different representation techniques.

To date the term "adequacy" has mainly been used in the context of higher-order abstract syntax techniques. However, none of this work has provided a clear, general definition of adequacy that is applicable to any representation technique; rather, particular instances of correspondences between object-languages and higher-order abstract syntax representations are typically called adequacy theorems. This has led to confusion between researchers familiar with different techniques, since properties such as "compositional bijection" that are called adequacy theorems in (for example) LF [10–12, 18] are couched in terms of LF or higher-order abstract syntax and bear little superficial resemblance to correctness properties established for other techniques [7, 8, 17, 20].

Because adequacy proofs for higher-order abstract syntax (like most proofs involving informal reasoning about languages with name-binding) are seldom published in full detail, some researchers (including the authors) have remained unpersuaded that these informal proofs have uncovered and addressed all pertinent issues. Conversely, some researchers [6] have criticized other techniques, suggesting that their correctness may be in doubt because properties resembling LF adequacy theorems have not been proved for them. As a matter of public record, such doubts have remained speculative in nature and, to date, no concrete shortcomings have been identified or articulated. Moreover, discussion of these problems has been

inconclusive, in part because there is no general, representation-independent definition of adequacy that is widely accepted.

We wish to distinguish two possible senses of the term "adequacy" which appear to have been conflated. By *informal adequacy* we mean that the formalization of a mathematical object matches a person's informal understanding of the object. Informal mathematical concepts are sometimes ambiguous or subjective, so informal adequacy is in general a subjective judgment. Informal adequacy is neither provable nor falsifiable by formal means: we cannot formally prove a relationship between an informal notion and a proposed formalization without introducing some new formalism whose (informal) adequacy could also be disputed. Informal adequacy is a rhetorical claim, outside the scope of mathematical reasoning.

Conversely, we use the term *formal adequacy* to describe a formalized relationship between two candidate formalizations of an object language (e.g., isomorphism with respect to a precisely defined mathematical structure). Formal adequacy cannot guarantee informal adequacy. However, it is at least reassuring if we can prove that a novel representation is equivalent to a well-understood, conventional one. Over time, we may thereby establish that many distinct formal presentations of a single informal (but important) object, such as the lambda-calculus, are all mutually equivalent [17]. The more formalizations that are equivalent, the more disconcerting it is if one is not; even if only two candidate formalizations exist, the absence of a provable link connecting them is cause for concern. Formal adequacy is not a matter of speculation but a mathematical claim that is subject to proof or disproof: in principle, it can be analyzed within a machine-checked logic. We will concentrate on formalizing adequacy in this sense and henceforth, we will use the term "adequacy" only in the formal sense.

1.1 Prior Work

In higher-order abstract syntax (HOAS), as supported in the LF [10] or higher-order logic programming [14] paradigms, a representation is usually considered adequate with respect to an object language (equipped with a suitable notion of substitution) provided there exists a *bijection* from the object-language terms to meta-language terms of an appropriate type that is *compositional* in the sense that the object-language substitution maps to metalanguage substitution. Proving adequacy properties for higher-order representations requires first establishing that meta-language terms have unique β-normal, η-long canonical forms, and once this is done, showing that a given encoding function is bijective and compositional. The necessary canonicalization properties for LF were studied by Harper and Pfenning [12]. The literature contains many proof sketches and some detailed proofs of ad hoc adequacy theorems [10–12, 18] for various object languages. Gardner [9] investigated the problem of relating object-language terms and logical derivations with their LF representations more systematically using indexed isomorphisms. To our knowledge none of these adequacy proofs for higher-order abstract syntax have been mechanically formalized (aside from the partial formalization by Urban et al. [23]).

The most rigorous treatments of adequacy for HOAS are due to Harper and Pfenning [12, Section 7] and Harper and Licata [11, Section 3]. Urban et al. [23]

formalized much of Harper and Pfenning's development in Nominal Isabelle/HOL, including the simple adequacy results sketched in Section 7 of that article. Our proof draws in part on ideas in that work, but we define encodings via functions rather than relations, and we investigate adequacy for judgments as well as the object language syntax. On the other hand, we focus on a simply-typed HOAS language rather than full dependently-typed LF. Harper and Licata [11] pursue an alternative approach to adequacy using a system called *Canonical LF* in which every expression is maintained in β-normal, η-long (or *canonical*) form. This is believed to ease reasoning about adequacy because we no longer need to reason modulo $\beta\eta$-equivalence; instead, we can inspect the structure of canonical forms. However, this canonical-forms invariant is maintained using an auxiliary notion, called *hereditary substitution* (introduced by Watkins et al. [24]). As discussed further below, hereditary substitution cannot be formalized easily within most proof assistants. To our knowledge, the key properties of Canonical LF have never been mechanically checked because of this complication.

There appears to be little agreement as to how to define adequacy for other representations, possibly because of the absence of a clear and general definition of adequacy for higher-order abstract syntax itself. There are some treatments of correctness properties for other techniques besides LF, notably Crole's [7] development of adequacy for the Hybrid system, Norrish and Vestergaard's [17] formalization of isomorphisms among nominal and de Bruijn representations of the untyped λ-calculus, and Urban's isomorphism between raw and nominal representations of the λ-calculus [21, Section 3]. Cheney and Urban [3, Section 5] also discuss adequacy in the context of nominal logic programming. As far as we know, of these only Urban's and Norrish and Vestergaard's work has been mechanically formalized.

1.2 Our Approach

We argue that the purpose of adequacy is to legitimize reasoning about one (mathematical or formal) structure by reasoning about another. Standard model-theoretic notions such as *isomorphism* or *interpretation* capture this notion in an abstract way [13]. In this article, we consider a class of adequacy properties based on interpretations among appropriate mathematical structures. This approach to adequacy generalizes that taken in the LF setting, and provides a clear recipe for studying adequacy that is largely independent of syntactic idiosyncrasies of the object and meta-languages, such as the treatment of variables, contexts, substitution and α-equivalence.

Our approach to formalizing adequacy for higher-order abstract syntax is based on explicitly establishing a canonical-forms theorem and then reasoning about the canonical forms, rather than maintaining canonical forms using hereditary substitution. While hereditary substitution appears to simplify adequacy proofs on paper, it cannot be defined as a primitive recursive function in (the current version of) Nominal Isabelle; instead, we need to define it as a relation. We have explored this alternative, and found that relational reasoning about substitution dramatically increases the amount of work needed for each proof. Many inference steps that can be handled automatically by equational simplification would have to be performed explicitly, one step at a time. There may be a way around this, but in this article we chose to focus on the better-understood approach based on ordinary substitution and

canonicalization, and leave formalizing Canonical LF and hereditary substitutions to future work.

Our approach to canonical forms also draws upon some prior work on canonical forms and equivalence checking in the λ-calculus. Coquand introduced an algorithm for testing $\beta\eta$-convertibility, along with a proof technique based on Kripke logical relations [4]. Crary [5] and Harper and Pfenning [12] give type-directed variants of Coquand's algorithm for testing convertibility in simple type theory and LF respectively. Canonical forms can then be extracted from algorithmic equivalence derivations. Correctness proofs for the type-directed algorithms have been formalized by Narboux and Urban [16] and Urban et al. [23] respectively. For our purposes, however, it is more convenient to prove soundness and completeness of a canonicalization algorithm directly (see Appendix).

Verifying adequacy does not seem inherently difficult, but is tedious and must currently be redone for each new representation. Instead of doing this on a case-by-case basis, we are investigating fundamental principles for systematically reasoning about adequacy. For the moment, the extant case studies can only suggest certain general principles. We leave their further investigation and codification to future work.

1.3 Summary

The contributions of this article are as follows:

1. We propose a clear, general, and formal characterization of adequacy as the existence of a model-theoretic interpretation [13] of the object-language within the meta-language. This definition is independent of inessential details of particular representation techniques and, in principle, applicable to any combination of object-language, representation technique, and formal framework.
2. We formalize an adequate interpretation of the untyped λ-calculus with reduction using a higher-order metalanguage, within Isabelle/HOL and the Nominal Datatype Package [21]. In doing so, we also prove the properties that are typically called "adequacy theorems" for such systems. To our knowledge, this is the first mechanized formalization and proof of such results for a higher-order abstract syntax representation.

This work extends the existing body of evidence that these and many other formalisms correctly capture the "real" λ-calculus (see Norrish and Vestergaard [17] for a more complete discussion of such results). Our formalization also includes reusable results such as the canonicalization theorem, and elucidates the patterns of reasoning that arise in adequacy proofs. This constitutes a library of useful results and a model proof of adequacy that may be extended or adapted to study adequacy for other languages, or as a starting point for improving support for automating such proofs.

We employ a metalanguage called λ^{HO}, a simply-typed λ-calculus that supports higher-order abstract syntax, and a logic called $\lambda^{\mathrm{HO}}_{\Rightarrow}$ (based on uniform, focused proof search in higher-order logic programming [15]) that can be used to define relations on higher-order representations (Section 2). We next introduce an object language, the (standard) untyped λ-calculus with β-reduction, and review its representation in λ^{HO}. Section 3 presents our approach to adequacy using concepts from model

theory, specifically *interpretations*. In Section 4, we show the central canonical-forms property for λ^{HO}, establishing that $\beta\eta$-equivalence classes are represented by unique canonical forms. We establish the main adequacy result in Section 5. In Section 5.1, we define the encoding function mapping object language terms to λ^{HO} terms, and establish the critical injectivity, surjectivity and compositionality properties of the encoding. Finally, in Section 5.2 we show that the encoding function preserves and reflects reduction. This is the most subtle aspect of the development and, to the best of our knowledge, comparable results have not previously been formalized within a machine-checked logic.

Every mathematical statement in this article except for Theorem 1 (a standard result of model theory, and not central to our development) has been formalized and proved using Nominal Isabelle. We follow the usual informal conventions concerning α-equivalence, but these are treated rigorously in the Nominal Isabelle formalization. In particular, to aid readability we have suppressed explicit mention of freshness and context-validity side-conditions whenever convenient. The full statements and proofs can be found in the formalization, which is available upon request.

2 Background

We take as our meta-language a simply-typed λ-calculus with constants, called λ^{HO}. The syntactic classes include expressions M, N, types τ, and contexts Γ generated by the following grammar:

$$M, N ::= c \mid x \mid M\ N \mid \Lambda x.N \qquad \tau ::= a \mid \tau \to \tau' \qquad \Gamma ::= \bullet \mid \Gamma, x{:}\tau$$

where x denotes one of a fixed infinite set of variables, c denotes one of a fixed set of constants, and a denotes one of a set of base types. To minimize confusion, we write $\Lambda x.M$ for meta-language λ-abstraction.

The (standard) definitions of the free variables function $FV(-)$, well-formedness judgment $\Gamma \vdash M : \tau$, and substitution operation $M\{N/x\}$ are shown in Fig. 1. The typing judgment $\Gamma \vdash M : \tau$ is implicitly parameterized by a signature Σ consisting of bindings $c : \tau$ of constants to their types, such that for each c there is at most one τ such that $c : \tau \in \Sigma$. Similarly, we restrict attention to contexts Γ that bind each variable at most once.

$$\begin{aligned} FV(x) &= \{x\} \\ FV(M\ N) &= FV(M) \cup FV(N) \\ FV(\Lambda x.M) &= FV(M) - \{x\} \end{aligned} \qquad \begin{aligned} x\{N/x\} &= N \\ y\{N/x\} &= y \qquad (x \neq y) \\ (M_1\ M_2)\{N/x\} &= M_1\{N/x\}\ M_2\{N/x\} \\ (\Lambda y.M)\{N/x\} &= \Lambda y.(M\{N/x\}) \qquad (y \neq x, y \notin FV(N)) \end{aligned}$$

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau} \qquad \frac{c : \tau \in \Sigma}{\Gamma \vdash c : \tau} \qquad \frac{\Gamma \vdash M : \tau_1 \to \tau_2 \quad \Gamma \vdash N : \tau_1}{\Gamma \vdash M\ N : \tau_2} \qquad \frac{\Gamma, x : \tau_1 \vdash M : \tau_2}{\Gamma \vdash \Lambda x.M : \tau_1 \to \tau_2}$$

Fig. 1 The meta-language λ^{HO}: a simply-typed λ-calculus with constants

The typing judgment satisfies standard weakening and substitution properties:

Proposition 1 (Properties of typing)

1. *(Weakening) If* $\Gamma \subseteq \Gamma'$ *and* $\Gamma \vdash M : \tau$ *then* $\Gamma' \vdash M : \tau$.
2. *(Substitution) If* $\Gamma, x{:}\tau' \vdash M : \tau$ *and* $\Gamma \vdash N : \tau'$ *then* $\Gamma \vdash M\{N/x\} : \tau$.

2.1 Definitional Equivalence

Meta-language expressions are considered equivalent modulo β and η-conversion. To be precise, we use a typed *definitional equivalence* judgment $\Gamma \vdash M = N : \tau$ (Fig. 2) to identify when two expressions are $\beta\eta$-convertible in a given context. Definitional equivalence is only derivable for well-typed terms, and satisfies weakening, substitution, and other standard properties:

Proposition 2 (Properties of definitional equivalence)

1. *(Validity) If* $\Gamma \vdash M = N : \tau$ *then* $\Gamma \vdash M : \tau$ *and* $\Gamma \vdash N : \tau$.
2. *(Reflexivity) If* $\Gamma \vdash M : \tau$ *then* $\Gamma \vdash M = M : \tau$.
3. *(Weakening) If* $\Gamma \subseteq \Gamma'$ *and* $\Gamma \vdash M = N : \tau$ *then* $\Gamma' \vdash M = N : \tau$.
4. *(Substitution) If* $\Gamma, x{:}\tau' \vdash M = M' : \tau$ *and* $\Gamma \vdash N : \tau'$ *then* $\Gamma \vdash M\{N/x\} = M'\{N/x\} : \tau$.
5. *(Functionality) If* $\Gamma, x{:}\tau' \vdash M = M' : \tau$ *and* $\Gamma \vdash N = N' : \tau'$ *then* $\Gamma \vdash M\{N/x\} = M'\{N'/x\} : \tau$.

2.2 A Higher-order Meta-logic

Besides the meta-language itself, we employ a simple logic that can be used to define relations over meta-language terms. We introduce a type constant o standing for the type of propositions. Propositions ϕ, ψ are expressions of the following forms:

$$\phi, \psi ::= A \mid \phi \supset \psi \mid \Pi x{:}\tau.\phi$$

Here, A is an object-language term that is being used as an atomic formula, and we use the notation $\phi \supset \psi$ for implication and $\Pi x{:}\tau.\phi$ for universal quantification in

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x = x : \tau}\ (=_{\text{var}}) \qquad \frac{c : \tau \in \Sigma}{\Gamma \vdash c = c : \tau}\ (=_{\text{cst}})$$

$$\frac{\Gamma \vdash M_1 = N_1 : \tau_1 \to \tau_2 \quad \Gamma \vdash M_2 = N_2 : \tau_1}{\Gamma \vdash M_1\ M_2 = N_1\ N_2 : \tau_2}\ (=_{\text{app}}) \qquad \frac{\Gamma, x : \tau_1 \vdash M = N : \tau_2}{\Gamma \vdash \Lambda x.M = \Lambda x.N : \tau_1 \to \tau_2}\ (=_{\text{lam}})$$

$$\frac{\Gamma \vdash M : \tau \to \tau' \quad x \notin FV(\Gamma, M)}{\Gamma \vdash \Lambda x.M\ x = M : \tau \to \tau'}\ (=_{\eta}) \qquad \frac{\Gamma, x{:}\tau' \vdash M : \tau \quad \Gamma \vdash N : \tau'}{\Gamma \vdash (\Lambda x.M)\ N = M\{N/x\} : \tau}\ (=_{\beta})$$

$$\frac{\Gamma \vdash N = M : \tau}{\Gamma \vdash M = N : \tau}\ (=_{\text{sym}}) \qquad \frac{\Gamma \vdash M = N : \tau \quad \Gamma \vdash N = K : \tau}{\Gamma \vdash M = K : \tau}\ (=_{\text{trans}})$$

Fig. 2 Definitional equivalence for λ^{HO}

$$\frac{\Gamma \vdash A : o}{\Gamma \vdash A \ \mathbf{prop}} \qquad \frac{\Gamma \vdash \phi \ \mathbf{prop} \quad \Gamma \vdash \psi \ \mathbf{prop}}{\Gamma \vdash \phi \supset \psi \ \mathbf{prop}} \qquad \frac{\Gamma, x{:}\tau \vdash \phi \ \mathbf{prop}}{\Gamma \vdash \Pi x{:}\tau.\phi \ \mathbf{prop}}$$

Fig. 3 Well-formed formulas of the meta-logic $\lambda^{\mathrm{HO}}_{\Rightarrow}$

$\lambda^{\mathrm{HO}}_{\Rightarrow}$. To simplify matters, we will not consider λ-abstraction or $\beta\eta$-conversion at the level of propositions.

We introduce a simple meta-logic that can be used to define relations over λ^{HO} terms. The well-formed formulas in a context Γ are defined in Fig. 3. We consider *theories* Φ to be sets of formulas. The judgments are $\Gamma; \Phi \Longrightarrow \phi$, which says that ϕ is derivable in context Γ using theory Φ, and $\Gamma; \Phi \mid \phi \longrightarrow A$, which says that atomic formula A is immediately derivable from ϕ in context Γ under assumptions Φ. The rules for derivability are given in Fig. 4. This system allows relations to be defined as simple higher-order logic programs. It is based on systems for uniform and focused proofs that have been investigated for higher-order logic programming [15]. Essentially, the idea is that we start with a cut-free, Gentzen-style sequent calculus. We then require right-rules to be applied first whenever the conclusion is non-atomic. When the conclusion is atomic, we select a hypothesis $\phi \in \Phi$ and proceed by applying left-rules to the focused formula ϕ to break it down into subgoals.

We have chosen to take the cut-free, uniform, focused proof search system as primitive rather than devote additional effort to formalizing cut-elimination and proof normalization results. These results are interesting and appear nontrivial to formalize, but they are generally well-understood and this issue seems orthogonal to the goals of this article. LF signatures can also be viewed as a form of higher-order logic programs, and this meta-logic is also closely related to canonical derivations in LF, so we believe that adequacy proofs based on this system could be adapted to handle LF as well.

Proposition 3 (Properties of well-formed formulas)

1. *If* $\Gamma \subseteq \Gamma'$ *and* $\Gamma \vdash \phi$ **prop** *then* $\Gamma' \vdash \phi$ **prop**.
2. *If* $\Gamma; \Phi \Longrightarrow A$ *and* $\Gamma \vdash A = A' : o$ *then* $\Gamma; \Phi \Longrightarrow A'$.
3. *If* $\Gamma; \Phi \mid \phi \longrightarrow A$ *and* $\Gamma \vdash A = A' : o$ *then* $\Gamma; \Phi \mid \phi \longrightarrow A'$.

We can also establish that $\lambda^{\mathrm{HO}}_{\Rightarrow}$ is closed under definitional equivalence of formulas and theories (suitably defined), but we do not need this result in the rest of the article.

$$\frac{\phi \in \Phi \quad \Gamma; \Phi \mid \phi \longrightarrow A}{\Gamma; \Phi \Longrightarrow A}\ (\mathrm{sel}) \qquad \frac{\Gamma \vdash A = A' : o}{\Gamma; \Phi \mid A' \longrightarrow A}\ (\mathrm{hyp})$$

$$\frac{\Gamma; \Phi, \phi \Longrightarrow \psi}{\Gamma; \Phi \Longrightarrow \phi \supset \psi}\ (\supset R) \qquad \frac{\Gamma; \Phi \Longrightarrow \phi \quad \Gamma; \Phi \mid \psi \longrightarrow A}{\Gamma; \Phi \mid \phi \supset \psi \longrightarrow A}\ (\supset L)$$

$$\frac{\Gamma, x{:}\tau; \Phi \Longrightarrow \phi}{\Gamma; \Phi \Longrightarrow \Pi x{:}\tau.\phi}\ (\Pi R) \qquad \frac{\Gamma \vdash M : \tau \quad \Gamma; \Phi \mid \phi\{M/x\} \longrightarrow A}{\Gamma; \Phi \mid \Pi x{:}\tau.\phi \longrightarrow A}\ (\Pi L)$$

Fig. 4 Derivability in the meta-logic $\lambda^{\mathrm{HO}}_{\Rightarrow}$

$$fv(x) = \{x\} \qquad x[u/x] = u$$

$$fv(t\ u) = fv(t) \cup fv(u) \qquad y[u/x] = y \quad (x \neq y)$$

$$fv(\lambda x.t) = fv(t) - \{x\} \qquad (t_1\ t_2)[u/x] = t_1[u/x]\ t_2[u/x]$$

$$(\lambda y.t)[u/x] = \lambda y.(t[u/x]) \quad (y \neq x, y \notin fv(u))$$

$$\frac{t \longrightarrow_\beta t'}{t\ u \longrightarrow_\beta t'\ u} \qquad \frac{u \longrightarrow_\beta u'}{t\ u \longrightarrow_\beta t\ u'} \qquad \frac{t \longrightarrow_\beta u}{\lambda x.t \longrightarrow_\beta \lambda x.u} \qquad \frac{}{(\lambda x.t)\ u \longrightarrow_\beta t[u/x]}$$

Fig. 5 The object-language $\mathcal{L}$: untyped λ-calculus with β-reduction

2.3 The Object Language, and its Representation

We take the untyped λ-calculus with β-reduction as our object language. We consider terms t, u generated by the grammar:

$$t ::= x \mid t\ u \mid \lambda x.t$$

We assume familiarity with the (standard) definitions of the free variables function $fv(-)$, the substitution operation $t[u/x]$, and the β-reduction relation shown in Fig. 5. We write $\mathcal{L}$ for the set of closed untyped λ-terms.

In Fig. 6, we show the representation of the object language in λ^{HO}, defined using a signature $\Sigma_{\mathcal{L}}$ defining constants for λ-abstraction and application, and a theory $\Phi_{\mathcal{L}}$ axiomatizing β-reduction on meta-language terms. This representation is essentially the standard higher-order representation of this object language afforded by LF or λProlog; it differs from typical presentations in these systems only in syntactic details. For example, we write the rule for β-reduction under a λ-abstraction as:

$$\Pi M, N : \mathsf{exp} \to \mathsf{exp}.\ \left(\Pi x : \mathsf{exp}.M\ x \dashrightarrow_\beta N\ x\right) \supset \mathsf{lam}\ M \dashrightarrow_\beta \mathsf{lam}\ N$$

and the same rule would be written in Twelf or λProlog concrete syntax as:

```
% Twelf
red_lam : red (lam M) (lam N) <- {x} red (M x) (N x).

% lambdaProlog
red (lam M) (lam N) :- pi x. red (M x) (N x).
```

Type constants: exp
Term constants: lam, app, $\dashrightarrow_\beta$

$$\Sigma_{\mathcal{L}} = \{\mathsf{lam} : (\mathsf{exp} \to \mathsf{exp}) \to \mathsf{exp}, \mathsf{app} : \mathsf{exp} \to \mathsf{exp} \to \mathsf{exp}, \dashrightarrow_\beta : \mathsf{exp} \to \mathsf{exp} \to o\}$$

$$\Phi_{\mathcal{L}} = \left\{ \begin{array}{l} (\mathrm{red\_app}_1) = \Pi M, N, P : \mathsf{exp}.M \dashrightarrow_\beta P \supset \mathsf{app}\ M\ N \dashrightarrow_\beta \mathsf{app}\ P\ N, \\ (\mathrm{red\_app}_2) = \Pi M, N, P : \mathsf{exp}.N \dashrightarrow_\beta P \supset \mathsf{app}\ M\ N \dashrightarrow_\beta \mathsf{app}\ M\ P, \\ (\mathrm{red\_lam}) = \Pi M, N : \mathsf{exp} \to \mathsf{exp}.(\Pi x : \mathsf{exp}.M\ x \dashrightarrow_\beta N\ x) \supset \mathsf{lam}\ M \dashrightarrow_\beta \mathsf{lam}\ N, \\ (\mathrm{red\_beta}) = \Pi M : \mathsf{exp} \to \mathsf{exp}.\Pi N : \mathsf{exp}.\mathsf{app}\ (\mathsf{lam}\ M)\ N \dashrightarrow_\beta M\ N \end{array} \right\}$$

Fig. 6 Signature and theory representing the object language

2.4 Formalization in Nominal Isabelle

In this article, we prefer to focus on the high-level structure of the proof, which we believe could in principle be redone in any sufficiently rich mechanized logic, and leave out most of the details of formalization that are specific to Nominal Isabelle. The translation from the informal mathematical notation used in the article and the formalization is straightforward, using the same principles illustrated in Urban [21] and Urban et al. [23]. In particular, we make free use of "nominal primitive recursion" and "strong induction principles" (described comprehensively in [20–22]) for nominal datatypes and inductive definitions without further comment, except to emphasize when such principles are especially useful (for example in avoiding variable capture). The reader interested in understanding the nuts and bolts of formalizing adequacy in Nominal Isabelle is invited to consult the formalization and related articles.

3 Adequacy and Interpretations

Adequacy formalizes an intuition that the object language and reduction relation are correctly represented by the higher-order signature and theory. What does this mean exactly? Much work on adequacy for LF asserts that adequacy means showing that the object language is isomorphic to its representation. However, this guideline does not explain why the isomorphism is needed or identify which mathematical structure the isomorphism should preserve. If, as we argued earlier, the purpose of adequacy is to legitimize reasoning about the object language that is actually performed on the meta-level representation, then adequacy should ensure that properties of the object-language correspond to properties of the meta-language.

For concreteness, we use the language of first-order logic and model theory [13] to talk about the object language, meta-language and their properties; other logics could also be considered. We recapitulate some standard elements of first-order model theory [13]. A first-order structure $\mathcal{A}$ over a relational signature $\Omega = \{R_1 : k_1, \ldots, R_n : k_n\}$ consists of a carrier set A together with an k-ary relation $R^{\mathcal{A}} \subseteq A^k$ for each k-ary relation symbol $R : k \in \Omega$. In what follows, we often abuse notation by writing $\mathcal{A}$ for both a structure and its carrier set. Two first-order structures over a common signature are *elementarily equivalent* if they satisfy the same first-order sentences. An *isomorphism* of first-order structures $\mathcal{A}, \mathcal{B}$ over the same signature Ω is a bijection $h : \mathcal{A} \to \mathcal{B}$ such that for $R^{\mathcal{A}}(a_1, \ldots, a_n) \iff R^{\mathcal{B}}(h(a_1), \ldots, h(a_n))$ for each $R : n \in \Omega$ and $a_1, \ldots, a_n \in \mathcal{A}$.

It is a standard result that isomorphic structures are elementarily equivalent [13]; hence, their properties transfer in both directions via the identity translation. However, elementarily equivalence does not imply isomorphism; for example, $\mathbb{Q}$ and $\mathbb{R}$ satisfy the same sentences over $<$. We argue that the essence of adequacy is the ability to translate properties of the object language to equivalent properties of the meta-language. Thus, existence of an isomorphism is a sufficient, but not necessary criterion for adequacy. Moreover, it is often inconvenient to work with isomorphisms of structures over a common signature, because doing so requires constructing a subset type and quotienting with respect to definitional equivalence. Working with equivalence classes and quotient constructions is often painful in a mechanical formalization (including in Nominal Isabelle). Hence, we prefer to work with a more

general notion called *interpretation* (see for example Hodges [13]). Intuitively, an interpretation is a relationship that shows that a structure of interest is isomorphic to a (definable) part of another structure, possibly modulo some (definable) equivalence relation. This implies, in particular, that any logically definable property of the object structure can be translated to an equivalent definable property of the representing structure.

We introduce a slightly different definition of interpretation than the standard one in model theory [13]. Since we are working in Isabelle/HOL, we will define interpretations in terms of HOL formulas and functions. We say that a HOL formula over structure $\mathcal{A}$ with signature Ω is one whose (free and bound) variables range over elements of $\mathcal{A}$ and whose atomic formulas are among those in Ω.

Definition 1 (Interpretation) Let $\mathcal{A}$ and $\mathcal{B}$ be first-order structures over relational signatures Ω and Ω' respectively. We say that $\mathcal{A}$ is interpretable in $\mathcal{B}$ if there exists:

1. A subset $\ulcorner\mathcal{A}\urcorner : \mathcal{B} \to \mathbf{bool}$ of $\mathcal{B}$.
2. A binary relation $\equiv^{\mathcal{A}}: \mathcal{B} \times \mathcal{B} \to \mathbf{bool}$ that is an equivalence relation on $\ulcorner\mathcal{A}\urcorner$.
3. For every n-ary relation symbol $R : n \in \Omega$, a definable n-ary predicate $\ulcorner R\urcorner : \mathcal{B}^n \to \mathbf{bool}$ over Ω'.
4. A function $\ulcorner -\urcorner : \mathcal{A} \to \ulcorner\mathcal{A}\urcorner$ such that
   (a) (Injectivity) For any $a, a' \in \mathcal{A}$, if $\ulcorner a\urcorner \equiv^{\mathcal{A}} \ulcorner a'\urcorner$ then $a = a'$.
   (b) (Surjectivity) If $b \in \ulcorner\mathcal{A}\urcorner$ then there exists $a \in \mathcal{A}$ such that $\ulcorner a\urcorner \equiv^{\mathcal{A}} b$.
   (c) (Preservation and Reflection) If $R \in \Omega$ then $(a_1, \ldots, a_n) \in R$ holds if and only if $(\ulcorner a_1\urcorner, \ldots, \ulcorner a_n\urcorner) \in \ulcorner R\urcorner$

If structure $\mathcal{A}$ over signature Ω is interpretable in structure $\mathcal{B}$ over signature Ω' via $\ulcorner -\urcorner$, we define the interpretation $\ulcorner P\urcorner$ of an Ω-formula P in $\mathcal{B}$ as:

$$\begin{array}{rlrl}
\ulcorner\top\urcorner &= \top & \ulcorner\bot\urcorner &= \bot \\
\ulcorner P \wedge Q\urcorner &= \ulcorner P\urcorner \wedge \ulcorner Q\urcorner & \ulcorner P \vee Q\urcorner &= \ulcorner P\urcorner \vee \ulcorner Q\urcorner \\
\ulcorner P \Rightarrow Q\urcorner &= \ulcorner P\urcorner \Rightarrow \ulcorner Q\urcorner & R(x_1, \ldots, x_n) &= \ulcorner R\urcorner(x_1, \ldots, x_n) \\
\ulcorner\forall x{:}\mathcal{A}.\, P\urcorner &= \forall x{:}\ulcorner\mathcal{A}\urcorner.\, \ulcorner P\urcorner & (&= \forall x{:}\mathcal{B}.x \in \ulcorner\mathcal{A}\urcorner \Rightarrow \ulcorner P\urcorner) \\
\ulcorner\exists x{:}\mathcal{A}.\, P\urcorner &= \exists x{:}\ulcorner\mathcal{A}\urcorner.\, \ulcorner P\urcorner & (&= \exists x{:}\mathcal{B}.x \in \ulcorner\mathcal{A}\urcorner \wedge \ulcorner P\urcorner)
\end{array}$$

Note in particular that this definition covers all cases since constants or function symbols of type $\mathcal{A}$ cannot be mentioned in P, and that $\ulcorner P\urcorner$ is a Ω'-formula. The key metatheorem about interpretability is that the preservation and reflection properties can be lifted to arbitrary (higher-order) formulas over Ω.

Theorem 1 (Interpretability* (Theorem 4.3.1 of [13])) *Suppose $\mathcal{A}$ is interpretable in $\mathcal{B}$. Let P be a formula over Ω whose variables are all of type $\mathcal{A}$. Then*

$$P(a_1, \ldots, a_n) \iff \ulcorner P\urcorner(\ulcorner a_1\urcorner, \ldots, \ulcorner a_n\urcorner)$$

We mark this theorem with an asterisk because we have not formalized it within Isabelle/HOL (unlike the other results in the article). Indeed, the Interpretability Theorem is a metatheoretic property about Isabelle/HOL formulas that cannot easily be stated and proved within Isabelle/HOL. However, the real point of this metatheorem is that the ingredients of the definition of interpretation provide what is

needed to prove any particular formula P equivalent to $\ulcorner P \urcorner$; in principle, this proof could be performed automatically for a given Isabelle/HOL formula P.

We will prove in Section 5 that:

Theorem 2 (Adequacy) *The structure $\mathcal{L}$ over signature $\{\longrightarrow_\beta : 2\}$, defined in Fig. 6, consisting of the set of closed untyped λ-terms with β-reduction, is interpretable as the set of closed λ^{HO} terms of type* exp *modulo definitional equivalence, with β-reduction defined via the $\lambda^{\mathrm{HO}}_{\Rightarrow}$ theory $\Phi_{\mathcal{L}}$.*

Note that this statement is different from the usual statement of adequacy theorems in LF (see for example [10–12, 18]): we restrict attention to closed terms and we do not mention syntactic properties such as compositionality or substitution-preservation explicitly. We have done so in order to provide a notion of adequacy that is clean, high-level, and independent of syntactic details (e.g. variables). The adequacy theorem above therefore only tells us that first-order properties of closed object language terms can be translated to equivalent properties of closed meta-language terms.

In fact, our results require showing analogous results for open terms, amounting to proving the existence of an indexed interpretation, analogous to the indexed isomorphisms used by Gardner [9]. It should be possible to extend Theorem 1 to a richer logic and thereby broaden the class of properties that can be transferred between object and meta-language. However, we prefer to focus on the familiar first-order setting.

We first present the components of the interpretation and sketch the proofs of parts (1–3) of the definition of interpretation at a high level. These have also been formalized, but we prefer not to burden the exposition with the relatively mundane details.

The interpretation of closed object language terms is the set of well-formed terms of type exp in the empty context (with respect to $\Sigma_{\mathcal{L}}$):

$$\ulcorner \mathcal{L} \urcorner \triangleq \{M \mid \bullet \vdash M : \mathsf{exp}\}$$

Likewise, the equivalence relation is definitional equivalence in the empty context (again with respect to $\Sigma_{\mathcal{L}}$):

$$(\equiv^{\mathcal{L}}) \triangleq \{(M, N) \mid \bullet \vdash M = N : \mathsf{exp}\}$$

This is easily seen to be an equivalence relation on $\ulcorner \mathcal{L} \urcorner$, by properties already established. Similarly, we will interpret the β-reduction relation $\longrightarrow_\beta$ on closed terms as follows:

$$(\longrightarrow^{\mathcal{L}}_{\beta}) \triangleq \left\{(M, N) \mid \bullet; \Phi_{\mathcal{L}} \Longrightarrow M \dashrightarrow_\beta N\right\}$$

The encoding function on elements of $\mathcal{L}$ is defined by primitive recursion over the nominal datatype of object λ-terms:

$$\ulcorner x \urcorner = x \qquad \ulcorner t\ u \urcorner = \mathsf{app}\ \ulcorner t \urcorner\ \ulcorner u \urcorner \qquad \ulcorner \lambda x.t \urcorner = \mathsf{lam}\ (\Lambda x.\ulcorner t \urcorner)$$

This function is defined for all object-language terms, open or closed. See Lemma 6 for the proof that the encoding function maps closed object terms to well-formed closed λ^{HO} terms.

In order to prove the injectivity, surjectivity, preservation and reflection properties, we will need to establish that λ^{HO} terms have unique canonical forms. We therefore turn to this next before finishing the proof of adequacy in Section 5.

4 Canonical Forms

We define canonical forms informally to be higher-order terms that are β-normalized and fully η-expanded—i.e., no η-expansions can be performed without introducing a β-redex. Canonical and atomic forms obey the following grammar rules:

$$M_c ::= M_a \mid \Lambda x.M_c \quad M_a ::= x \mid c \mid M_a\ M_c$$

We give rules for identifying canonical and atomic forms in Fig. 7. The judgment $\Gamma \vdash M \Uparrow \tau$ says that M is a canonical form of type τ in context Γ, and likewise $\Gamma \vdash M \downarrow \tau$ says that M is an atomic form of type τ with respect to Γ. Obviously, these rules are a subsystem of those in Fig. 1. A canonical form of base type is an atomic form of that type, whereas a canonical form of function type must be a Λ-abstraction whose body is a canonical form of the result type. For example, the canonical forms of type exp are of the form

$$M_{\mathcal{L}}, N_{\mathcal{L}} ::= x \mid \mathsf{app}\ M_{\mathcal{L}}\ N_{\mathcal{L}} \mid \mathsf{lam}\ (\Lambda x.M_{\mathcal{L}})$$

This seems obvious, but it requires some work to prove this formally (we will return to this point in Section 5).

In the rest of this section, we show that each well-formed λ^{HO} term is definitionally equivalent to a canonical form; moreover, if two λ^{HO} terms are definitionally equivalent then they have the same canonical form.

4.1 Weak Head Reduction

We will employ a (standard) weak head reduction relation $M \xrightarrow{\mathrm{whr}} N$ and weak head normal forms predicate $\mathrm{whnf}(M)$, shown in Figs. 8 and 9 respectively. Here we state and prove some properties of weak head reduction and weak head normal forms that will be needed frequently, including subject reduction.

Proposition 4 (Properties of weak head reduction)

1. *(Identity)* $(\Lambda x.M)\ x \xrightarrow{\mathrm{whr}} M$
2. *(Determinacy) If* $M \xrightarrow{\mathrm{whr}} M'$ *and* $M \xrightarrow{\mathrm{whr}} M''$ *then* $M' = M''$.
3. *(Substitution) If* $M \xrightarrow{\mathrm{whr}} M'$ *then* $M\{N/x\} \xrightarrow{\mathrm{whr}} M'\{N/x\}$.
4. *For any* M, *we have* $\mathrm{whnf}(M)$ *if and only if* $\nexists M'.M \xrightarrow{\mathrm{whr}} M'$.

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x \downarrow \tau} \qquad \frac{c : \tau \in \Sigma}{\Gamma \vdash c \downarrow \tau} \qquad \frac{\Gamma \vdash M \downarrow \tau_1 \to \tau_2 \quad \Gamma \vdash N \Uparrow \tau_1}{\Gamma \vdash M\ N \downarrow \tau_2}$$

$$\frac{\Gamma \vdash M \downarrow a}{\Gamma \vdash M \Uparrow a} \qquad \frac{\Gamma, x : \tau_1 \vdash M \Uparrow \tau_2}{\Gamma \vdash \lambda x.M \Uparrow \tau_1 \to \tau_2}$$

Fig. 7 Canonical and atomic forms

$$\frac{}{(\Lambda x.M)\ N \xrightarrow{\text{whr}} M\{N/x\}}\ (\text{whr}_\beta) \qquad \frac{M \xrightarrow{\text{whr}} M'}{M\ N \xrightarrow{\text{whr}} M'\ N}\ (\text{whr}_{\text{app}})$$

Fig. 8 Weak head reduction

Theorem 3 (Subject reduction) *Assume* $M \xrightarrow{\text{whr}} N$ *and* $\Gamma \vdash M : \tau$. *Then* $\Gamma \vdash M = N : \tau$ *and* $\Gamma \vdash N : \tau$.

Proof Structural induction on the derivation of $M \xrightarrow{\text{whr}} N$, and case analysis on the well-formedness derivation. □

4.2 Canonicalization

The judgments for canonicalization are shown in Fig. 10. These rules, broadly, canonicalize a (not necessarily well-formed) λ^{HO} term in a type-directed way by η-expansion at function type, followed by weak head normalization at base type. Once the term is in weak head normal form, we can use the auxiliary judgment $\Gamma \vdash M \downarrow N : \tau$ to match the top portion of the weak head normalized term to its atomic form, and recursively canonicalize the arguments.

We first establish some simple syntactic properties of canonicalization. Weakening is standard and straightforward:

Lemma 1 (Weakening of canonicalization) *Assume that* $\Gamma \subseteq \Gamma'$. *Then:*

1. *If* $\Gamma \vdash M \downarrow N : \tau$ *then* $\Gamma' \vdash M \downarrow N : \tau$.
2. *If* $\Gamma \vdash M \Uparrow N : \tau$ *then* $\Gamma' \vdash M \Uparrow N : \tau$.

Canonicalization is sound in the sense that it produces a canonical form of the appropriate type, and if we canonicalize a well-formed λ^{HO} term then its canonical form is definitionally equivalent:

Theorem 4 (Soundness)

1. *If* $\Gamma \vdash M \downarrow N : \tau$ *then* $\Gamma \vdash N \downarrow \tau$.
2. *If* $\Gamma \vdash M \Uparrow N : \tau$ *then* $\Gamma \vdash N \Uparrow \tau$.

In either case, if in addition $\Gamma \vdash M : \tau$ *then* $\Gamma \vdash M = N : \tau$.

Proof The first two parts are trivial by induction on derivations. For the last part, proof is by induction on the canonicalization derivation, using inversion on the typing derivation of M. In the case for rule ($\Uparrow_{\text{fun}}$) we need weakening and η-expansion. In the case for rule ($\Uparrow_{\text{whr}}$) we need the weak head subject reduction theorem. □

Fig. 9 Weak head normal forms

$$\frac{}{\text{neut}(x)}\quad \frac{}{\text{neut}(c)}\quad \frac{\text{neut}(M)}{\text{neut}(M\ N)}\quad \frac{\text{neut}(M)}{\text{whnf}(M)}\quad \frac{}{\text{whnf}(\Lambda x.M)}$$

$$\frac{x:\tau \in \Gamma}{\Gamma \vdash x \downarrow x:\tau}\ (\downarrow_{\text{var}}) \qquad \frac{c:\tau \in \Sigma}{\Gamma \vdash c \downarrow c:\tau}\ (\downarrow_{\text{cst}})$$

$$\frac{\Gamma \vdash M_1 \downarrow N_1 : \tau_1 \to \tau_2 \quad \Gamma \vdash M_2 \Uparrow N_2 : \tau_1}{\Gamma \vdash M_1\ M_2 \downarrow N_1\ N_2 : \tau_2}\ (\downarrow_{\text{app}}) \qquad \frac{\Gamma \vdash M \downarrow N : a}{\Gamma \vdash M \Uparrow N : a}\ (\Uparrow\downarrow)$$

$$\frac{\Gamma, x:\tau_1 \vdash M\ x \Uparrow N : \tau_2}{\Gamma \vdash M \Uparrow \Lambda x.N : \tau_1 \to \tau_2}\ (\Uparrow_{\text{fun}}) \qquad \frac{M \xrightarrow{\text{whr}} M' \quad \Gamma \vdash M' \Uparrow N : a}{\Gamma \vdash M \Uparrow N : a}\ (\Uparrow_{\text{whr}})$$

Fig. 10 Canonicalization

The following lemmas provide the needed foundation to show that canonicalization is deterministic (Theorem 5).

Lemma 2 *If* $\Gamma \vdash M \downarrow N : \tau$ *then* $\text{neut}(M)$. *Moreover, if* $\Gamma \vdash M \Uparrow N : a$ *and* $\text{whnf}(M)$ *then* $\Gamma \vdash M \downarrow N : a$.

Proof Both parts follow by straightforward simultaneous induction. For the second part, note that the only rule that can apply is $(\Uparrow\downarrow)$. The rule $(\Uparrow_{\text{fun}})$ cannot apply because the type is a base type a and $(\Uparrow_{\text{whr}})$ cannot apply because M is weak head normalized. □

Lemma 3 *If* $\Gamma \vdash M \Uparrow N : \tau$ *and* $M \xrightarrow{\text{whr}} M'$ *then* $\Gamma \vdash M' \Uparrow N : \tau$.

Proof By induction on the structure of derivations, using Lemma 2 and determinacy of weak head reduction. □

Lemma 4 *If* $\Gamma \vdash M \Uparrow N : \tau \to \tau'$ *and* $x \notin FV(\Gamma, M, N)$ *then there exists* K *with* $N = \Lambda x.K$ *and* $\Gamma, x{:}\tau \vdash M\ x \Uparrow K : \tau'$.

Proof Straightforward, by inversion and standard permutative renaming reasoning. □

Theorem 5 (Determinacy of canonicalization)

1. *If* $\Gamma \vdash M \downarrow N : \tau$ *and* $\Gamma \vdash M \downarrow N' : \tau'$ *then* $N = N'$ *and* $\tau = \tau'$.
2. *If* $\Gamma \vdash M \Uparrow N : \tau$ *and* $\Gamma \vdash M \Uparrow N' : \tau'$ *then* $N = N'$.

Proof By induction on derivations. Most cases are straightforward using inversion and the induction hypothesis. We need Lemma 2 in the case for $(\Uparrow\downarrow)$, Lemma 4 in the case for $(\Uparrow_{\text{fun}})$ and Lemma 3 in the case for $(\Uparrow_{\text{whr}})$. □

Moreover, canonicalization is idempotent in that a canonical form canonicalizes (only) to itself.

Lemma 5 (Idempotence)

1. *If* $\Gamma \vdash N \downarrow \tau$ *then* $\Gamma \vdash N \downarrow N : \tau$.
2. *If* $\Gamma \vdash N \Uparrow \tau$ *then* $\Gamma \vdash N \Uparrow N : \tau$.

Proof Straightforward by induction on the given derivations. □

The key property we need to reason about adequacy is that definitionally equivalent terms have unique canonical forms. We have already established that canonicalization is sound with respect to definitional equivalence and canonical forms are unique when they exist.

Theorem 6 (Completeness) *If* $\Gamma \vdash M = N : \tau$ *then for some* K*, we have* $\Gamma \vdash M \Uparrow K : \tau$ *and* $\Gamma \vdash N \Uparrow K : \tau$*. Moreover,* K *is the unique canonical form definitionally equivalent to* M *and* N*.*

In an Appendix, we prove Theorem 6 by a Kripke logical relations argument similar to that used by Crary [5] and (for LF) by Harper and Pfenning [12] to prove the completeness of type-directed algorithms for definitional equivalence. Both proofs have been formalized previously using Nominal Isabelle [16, 23], and our proof follows a similar strategy, but there are some significant differences: in particular, we prove the existence of a common canonical form for M and N directly, rather than by extracting it from an algorithmic equivalence derivation as is done in Harper and Pfenning [12]. However, the proof details are not needed in order to appreciate the applications of Theorem 6 to the adequacy proof.

5 The Proof of Adequacy

Before we begin proving the remaining properties needed to show that the encoding $\ulcorner - \urcorner$ is an interpretation, we need to introduce some additional concepts.

Although we have stated the adequacy theorem in terms of closed terms, in general we also need to relate open terms (with, say, free variables $x_1, \ldots, x_n$) to λ^{HO} contexts (with, say, free variables $x_1{:}\mathsf{exp}, \ldots, x_n{:}\mathsf{exp}$). To make this precise, we introduce untyped contexts X, which are lists of distinct variables, and define a translation on contexts as follows:

$$\ulcorner \bullet \urcorner = \bullet \qquad \ulcorner X, x \urcorner = \ulcorner X \urcorner, x{:}\mathsf{exp}$$

We also introduce a simple validity relation $X \vdash t$, meaning that X is a list of distinct variables that includes all of the free variables of t.

$$\frac{x \in X}{X \vdash x} \qquad \frac{X \vdash t \quad X \vdash u}{X \vdash t\ u} \qquad \frac{X, x \vdash t \quad x \notin X}{X \vdash \lambda x.t}$$

It is easy to show that $X \vdash t \iff fv(t) \subseteq X$, and that $\bullet \vdash t$ holds if and only if t is closed. In addition, we will need the following key property:

Lemma 6 *For any* X*, we have* $X \vdash t$ *if and only if* $\ulcorner X \urcorner \vdash \ulcorner t \urcorner : \mathsf{exp}$*.*

Proof Proof in the forward direction is by induction on the derivation of $X \vdash t$. In the reverse direction, the proof proceeds by induction on the structure of t and inversion on the typing derivation. □

Note in particular that the forward direction of Lemma 6 shows that $\ulcorner - \urcorner$ maps closed object terms to elements of $\ulcorner \mathcal{L} \urcorner$, as required by the definition of interpretation.

5.1 The Encoding is a Compositional Bijection

Injectivity To prove injectivity, we introduce an inductive, relational definition of the graph of the encoding as follows:

$$\frac{}{x \sim x} \qquad \frac{t \sim M \quad u \sim N}{t\ u \sim \mathsf{app}\ M\ N} \qquad \frac{t \sim M}{\lambda x.t \sim \mathsf{lam}\ (\Lambda x.M)}$$

Lemma 7 *For any t and M, we have $\ulcorner t \urcorner = M$ if and only if $t \sim M$.*

Proof The forward direction is by induction on t. The second part is by induction on the derivation of $t \sim M$. □

Lemma 8 (Injectivity on graph) *If $t \sim M$ and $u \sim M$ then $t = u$.*

Proof By induction on the derivation of $t \sim M$, and inversion on the derivation of $u \sim M$. □

Corollary 1 (Injectivity) *The function $\ulcorner - \urcorner$ is injective.*

Now, we can prove that the encoding satisfies part 4(a) of the definition of interpretation:

Theorem 7 (Injectivity on definitional equivalence classes) *If $\ulcorner X \urcorner \vdash \ulcorner t \urcorner = \ulcorner u \urcorner : \mathsf{exp}$ then $t = u$. In particular, if t, u are closed and $\bullet \vdash \ulcorner t \urcorner = \ulcorner u \urcorner : \mathsf{exp}$ then $t = u$.*

Proof By completeness, we know both $\ulcorner t \urcorner$ and $\ulcorner u \urcorner$ have a common canonical form K. Moreover since $\ulcorner t \urcorner$ and $\ulcorner u \urcorner$ are both themselves canonical, by determinacy (Theorem 5) and idempotence (Lemma 5) we have $\ulcorner t \urcorner = K = \ulcorner u \urcorner$. By injectivity of $\ulcorner - \urcorner$, we conclude $t = u$. □

Surjectivity We break the proof of surjectivity down into several stages. We define a predicate *range* as follows:

$$\frac{}{range(x)} \qquad \frac{range(M) \quad range(N)}{range(\mathsf{app}\ M\ N)} \qquad \frac{range(M)}{range(\mathsf{lam}\ (\Lambda x.M))}$$

We first prove $range(-)$ characterizes the range of $\ulcorner - \urcorner$. We will then show that the range predicate also defines the set of canonical forms. Finally, we combine this with the canonical forms theorem to show that every well-formed expression of appropriate type is definitionally equivalent to the encoding of an object term.

Lemma 9 *For any M, we have $range(M)$ if and only if there exists t such that $\ulcorner t \urcorner = M$.*

Next we need to show that the range predicate correctly identifies the canonical forms. To do this, we need to proceed by height induction on λ^{HO} terms and apply inversion principles to show that canonical forms must be of the same shape as the elements of $range(-)$.

Lemma 10 (Atomic forms have at most two arguments) *For any X, M, τ, if $\ulcorner X \urcorner \vdash M \downarrow \tau$ (with respect to $\Sigma_{\mathcal{L}}$) then τ is not of the form $\tau_1 \to (\tau_2 \to (\tau_3 \to \tau_4))$.*

Proof By height induction on M and inversion on derivations. In the base cases for constants and variables, the proof is by case analysis of the possible types. The inductive case involving application follows because in the application rule for atomic forms, the type in the hypothesis has one more argument than in the conclusion. □

Lemma 11 *If $\ulcorner X \urcorner \vdash M \Uparrow \mathsf{exp}$ then either:*

1. *M is a variable $x \in X$; or*
2. *There exist M_1, M_2 such that $M = \mathsf{app}\ M_1\ M_2$ and $\ulcorner X \urcorner \vdash M_1 \downarrow \mathsf{exp}$ and $\ulcorner X \urcorner \vdash M_2 \downarrow \mathsf{exp}$; or*
3. *There exist x, M' with $x \notin X$ and $M = \mathsf{lam}(\Lambda x.M')$ and $\ulcorner X, x \urcorner \vdash M' \downarrow \mathsf{exp}$.*

Proof Case analysis, using the previous lemma to cut off inversion. □

Lemma 10 is admittedly ugly, but there is no obvious way to avoid it: for any finite signature there will always be a finite bound on the number of arguments, but to prove a specific case analysis lemma such as Lemma 11 we need to know a constant upper bound to cut off the inversion.

Theorem 8 *For any X, M, we have $range(M)$ and $\ulcorner X \urcorner \vdash M : \mathsf{exp}$ if and only if $\ulcorner X \urcorner \vdash M \Uparrow \mathsf{exp}$.*

Proof The forward direction is immediate by induction on derivations of $range(M)$. For the reverse direction, we can use height induction on M, using the previous lemma to split into cases. □

Corollary 2 *If $\ulcorner X \urcorner \vdash M \Uparrow \mathsf{exp}$ then there exists t such that $X \vdash t$ and $\ulcorner t \urcorner = M$.*

Now we can show that the encoding satisfies the desired part 4(b) of the definition of interpretation.

Theorem 9 (Surjectivity) *If $\ulcorner X \urcorner \vdash M : \mathsf{exp}$ then there exists t such that $X \vdash t$ and $\ulcorner X \urcorner \vdash \ulcorner t \urcorner = M : \mathsf{exp}$. In particular if $\bullet \vdash M : \mathsf{exp}$ then there exists a closed t with $\bullet \vdash \ulcorner t \urcorner = M : \mathsf{exp}$.*

Bijectivity and Compositionality In treatments of adequacy for LF, it is often expected that the encoding function be a bijection mapping object terms to canonical forms, and in addition should be "compositional", which is generally taken to mean that it should satisfy the equation

$$\ulcorner t[u/x] \urcorner = \ulcorner t \urcorner \{\ulcorner u \urcorner / x\}$$

The bijectivity property follows immediately from Lemma 6, injectivity and surjectivity:

Corollary 3 *For any X, the function $\ulcorner - \urcorner$ is a bijection*

$$\ulcorner - \urcorner : \{t \mid X \vdash t\} \to \{M \mid \ulcorner X \urcorner \vdash M \Uparrow \mathsf{exp}\}$$

between the set of object language terms (over variables X) and the set of definitional equivalence classes of meta-language terms (represented by canonical forms) of type exp *(over context $\ulcorner X \urcorner$).*

In our formalization, the compositionality or substitution-preservation property is also straightforward by induction on the structure of terms. This is straightforward because variables in expressions in the range of the bijection are always of type exp, hence these substitutions never introduce any β-reductions.

Theorem 10 (Compositionality) *Let t, u be object terms and x be a variable. Then*

$$\ulcorner t[u/x] \urcorner = \ulcorner t \urcorner \{\ulcorner u \urcorner / x\}$$

Proof Induction on t, avoiding variable capture in x and u. □

Compositionality does not correspond to any one part of the definition of interpretation that we are using, since we are only proving that relations on closed object terms are interpreted as other relations on closed λ^{HO} terms. However, we still need to establish compositionality in order to prove that the encoding preserves and reflects β-reduction.

5.2 The Encoding Preserves and Reflects Reduction

Ultimately, we want to show that for closed terms t, u, we have $t \longrightarrow_\beta u$ if and only if $\bullet; \Phi_{\mathcal{L}} \Longrightarrow \ulcorner t \urcorner \dashrightarrow_\beta \ulcorner u \urcorner$ is derivable. To show this we will need to prove more general versions of these statements involving open terms. We refer to the forward, easier direction as *preservation*, and the reverse direction as *reflection*.

Preservation We first establish that the rules of object language β-equivalence are correctly simulated by derivability in $\Phi_{\mathcal{L}}$:

Lemma 12 (Rule preservation)

1. *If* $\Gamma; \Phi_{\mathcal{L}} \Longrightarrow M \dashrightarrow_\beta P$ *and* $\Gamma \vdash M, N, P : \mathsf{exp}$ *then* $\Gamma; \Phi_{\mathcal{L}} \Longrightarrow \mathsf{app}\ M\ N \dashrightarrow_\beta \mathsf{app}\ P\ N$.
2. *If* $\Gamma; \Phi_{\mathcal{L}} \Longrightarrow N \dashrightarrow_\beta P$ *and* $\Gamma \vdash M, N, P : \mathsf{exp}$ *then* $\Gamma; \Phi_{\mathcal{L}} \Longrightarrow \mathsf{app}\ M\ N \dashrightarrow_\beta \mathsf{app}\ M\ P$.
3. *If* $\Gamma, x{:}\mathsf{exp}; \Phi_{\mathcal{L}} \Longrightarrow M \dashrightarrow_\beta M'$ *and* $\Gamma, x{:}\mathsf{exp} \vdash M, M' : \mathsf{exp}$ *and* $x \notin FV(\Gamma)$ *then* $\Gamma; \Phi_{\mathcal{L}} \Longrightarrow \mathsf{lam}\ (\Lambda x.M) \dashrightarrow_\beta \mathsf{lam}\ (\Lambda x.M')$.
4. *If* $\Gamma, x{:}\mathsf{exp} \vdash M : \mathsf{exp}$ *and* $\Gamma \vdash N : \mathsf{exp}$ *and* $x \notin FV(\Gamma, N)$ *then* $\Gamma; \Phi_{\mathcal{L}} \Longrightarrow \mathsf{app}\ (\mathsf{lam}\ (\Lambda x.M))\ N \dashrightarrow_\beta M\{N/x\}$.

Proof In each case, we proceed by looking up the appropriate rule in $\Phi_{\mathcal{L}}$, renaming the quantified variables to fresh names, and instantiating.

- The cases for application-congruences are straightforward (albeit tedious). One case is shown in Fig. 11a; the other is symmetric.
- In the rule for λ-congruence we need to show that the meta-logic's use of the Π-quantifier is correct. This requires β-conversion and so is more involved. First, let x' be a fresh variable name. By substitution we can derive $\Gamma, x'{:}\mathsf{exp}; \Phi_{\mathcal{L}} \Longrightarrow M\{x'/x\} \dashrightarrow_\beta M'\{x'/x\}$. Thus, by β-converting both sides, using Proposition 3, we have $\Gamma, x'{:}\mathsf{exp}; \Phi_{\mathcal{L}} \Longrightarrow (\Lambda x.M)\ x' \dashrightarrow_\beta (\Lambda x.M')\ x'$. We can then reason as shown in Fig. 11b.
- In the case for β-reduction we need to verify that the substitution in the β-reduction step is simulated by β-reduction in the meta-language. The derivation is shown in Fig. 11c. Deriving $\Gamma \vdash \mathsf{app}\ (\mathsf{lam}\ (\Lambda x.M))\ N \dashrightarrow_\beta (\Lambda x.M)\ N = \mathsf{app}\ (\mathsf{lam}\ (\Lambda x.M))\ N \dashrightarrow_\beta M\{N/x\} : o$ requires a number of further inference steps. □

Theorem 11 (Encoding preserves reduction) *Assume $X \vdash t, u$. Then if $t \longrightarrow_\beta u$ then $\ulcorner X \urcorner; \Phi_{\mathcal{L}} \Longrightarrow \ulcorner t \urcorner \dashrightarrow_\beta \ulcorner u \urcorner$.*

Proof By induction on the derivation of $t \longrightarrow_\beta u$. In each case we employ the corresponding derivable rule from the previous lemma. In the case for β-reduction steps we need Theorem 10. □

So in particular we have established the preservation direction of part 4(c) of the definition of interpretation:

Corollary 4 *If t, u are closed terms and $t \longrightarrow_\beta u$ then $\bullet; \Phi_{\mathcal{L}} \Longrightarrow \ulcorner t \urcorner \dashrightarrow_\beta \ulcorner u \urcorner$.*

Reflection As might be expected, the reflection direction is harder, and its formalization involves some subtle choices concerning how to formulate inversion principles and when to appeal to canonical-forms principles.

Informally, the proof idea is to proceed by inspecting the possible derivations of $\bullet; \Phi_{\mathcal{L}} \Longrightarrow \ulcorner t \urcorner \dashrightarrow_\beta \ulcorner u \urcorner$ to establish that $t \longrightarrow_\beta u$. Given such a derivation, we wish to proceed by applying inversion rules, examining all cases to see that any derivation in $\lambda^{\mathrm{HO}}_{\Rightarrow}$ must end with a pattern of reasoning similar to those exhibited in the proof of Lemma 12, then appealing to the induction hypothesis. However, while this informal strategy appears straightforward or even trivial, mechanically formalizing it is a significant engineering challenge. There are several reasons for this.

First, we cannot proceed directly by induction on the structure of derivations of $\bullet; \Phi_{\mathcal{L}} \Longrightarrow \ulcorner t \urcorner \dashrightarrow_\beta \ulcorner u \urcorner$, because each single rule in the object language corresponds to many rule applications in the meta-logic. Instead, we need to reason by height-induction on the $\lambda^{\mathrm{HO}}_{\Rightarrow}$ derivations.

Second, it is also challenging, purely as a practical matter, to prove inversion principles for each formula in $\Phi_{\mathcal{L}}$. The proofs of these inversion principles involve reasoning about relatively large terms, compared with the rest of the development. For example, the abstract syntax tree for the smallest rule (red_beta) contains 19 nodes.

(a)

$$\dfrac{\dfrac{}{\Gamma;\, \Phi_{\mathcal{L}} \Longrightarrow M \dashrightarrow_\beta P} \quad \dfrac{}{\Gamma;\, \Phi_{\mathcal{L}} \mid \mathsf{app}\, M\, N \dashrightarrow_\beta \mathsf{app}\, P\, N \longrightarrow \mathsf{app}\, M\, N \dashrightarrow_\beta \mathsf{app}\, P\, N}}{\dfrac{\Gamma \vdash P : \mathsf{exp} \quad \dfrac{\Gamma;\, \Phi_{\mathcal{L}} \mid M \dashrightarrow_\beta P \supset \mathsf{app}\, M\, N \dashrightarrow_\beta \mathsf{app}\, P\, N \longrightarrow \mathsf{app}\, M\, N \dashrightarrow_\beta \mathsf{app}\, P\, N}{\Gamma;\, \Phi_{\mathcal{L}} \mid \Pi P : \mathsf{exp}.M \dashrightarrow_\beta P \supset \mathsf{app}\, M\, N \dashrightarrow_\beta \mathsf{app}\, P\, N \longrightarrow \mathsf{app}\, M\, N \dashrightarrow_\beta \mathsf{app}\, P\, N}}{\dfrac{\Gamma \vdash N : \mathsf{exp} \quad \dfrac{\Gamma;\, \Phi_{\mathcal{L}} \mid \Pi N, P : \mathsf{exp}.M \dashrightarrow_\beta P \supset \mathsf{app}\, M\, N \dashrightarrow_\beta \mathsf{app}\, P\, N \longrightarrow \mathsf{app}\, M\, N \dashrightarrow_\beta \mathsf{app}\, P\, N}{\Gamma;\, \Phi_{\mathcal{L}} \mid \Pi M, N, P : \mathsf{exp}.M \dashrightarrow_\beta P \supset \mathsf{app}\, M\, N \dashrightarrow_\beta \mathsf{app}\, P\, N \longrightarrow \mathsf{app}\, M\, N \dashrightarrow_\beta \mathsf{app}\, P\, N}}{\Gamma;\, \Phi_{\mathcal{L}} \Longrightarrow \mathsf{app}\, M\, N \dashrightarrow_\beta \mathsf{app}\, P\, N}}{\Gamma \vdash M : \mathsf{exp}}$$

(b)

$$\dfrac{\dfrac{}{\Gamma, x' : \mathsf{exp};\, \Phi_{\mathcal{L}} \Longrightarrow (\Lambda x.M)\, x' \dashrightarrow_\beta (\Lambda x.M')\, x'} \quad \dfrac{}{\Gamma;\, \Phi_{\mathcal{L}} \mid \mathsf{lam}\, (\Lambda x.M) \dashrightarrow_\beta \mathsf{lam}\, (\Lambda x.M') \longrightarrow \mathsf{lam}(\Lambda x.M) \dashrightarrow_\beta \mathsf{lam}\, (\Lambda x.M')}}{\dfrac{\Gamma;\, \Phi_{\mathcal{L}} \Longrightarrow \Pi x' : \mathsf{exp}.(\Lambda x.M)\, x' \dashrightarrow_\beta (\Lambda x.M')\, x' \quad \dfrac{\Gamma;\, \Phi_{\mathcal{L}} \mid (\Pi x' : \mathsf{exp}.(\Lambda x.M)\, x' \dashrightarrow_\beta (\Lambda x.M')\, x') \supset \mathsf{lam}\, (\Lambda x.M) \dashrightarrow_\beta \mathsf{lam}\, (\Lambda x.M') \longrightarrow \mathsf{lam}(\Lambda x.M) \dashrightarrow_\beta \mathsf{lam}\, (\Lambda x.M')}{\Gamma;\, \Phi_{\mathcal{L}} \mid \Pi N : \mathsf{exp} \to \mathsf{exp}.(\Pi x' : \mathsf{exp}.(\Lambda x.M)\, x' \dashrightarrow_\beta N\, x') \supset \mathsf{lam}\, (\Lambda x.M) \dashrightarrow_\beta \mathsf{lam}\, N \longrightarrow \mathsf{lam}(\Lambda x.M) \dashrightarrow_\beta \mathsf{lam}\, (\Lambda x.M')}}{\dfrac{\Gamma;\, \Phi_{\mathcal{L}} \mid \Pi M, N : \mathsf{exp} \to \mathsf{exp}.(\Pi x' : \mathsf{exp}.M\, x' \dashrightarrow_\beta N\, x') \supset \mathsf{lam}\, M \dashrightarrow_\beta \mathsf{lam}\, N \longrightarrow \mathsf{lam}(\Lambda x.M) \dashrightarrow_\beta \mathsf{lam}\, (\Lambda x.M')}{\Gamma;\, \Phi_{\mathcal{L}} \Longrightarrow \mathsf{lam}(\Lambda x.M) \dashrightarrow_\beta \mathsf{lam}\, (\Lambda x.M')}}{\Gamma;\, \Phi_{\mathcal{L}} \Longrightarrow (\Lambda x.M)\, x' \dashrightarrow_\beta (\Lambda x.M')\, x'}$$

(c)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \mathsf{app}\, (\mathsf{lam}\, (\Lambda x.M))\, N \dashrightarrow_\beta (\Lambda x.M)\, N = \mathsf{app}\, (\mathsf{lam}\, (\Lambda x.M))\, N \dashrightarrow_\beta M\{N/x\} : o}}{\Gamma;\, \Phi_{\mathcal{L}} \mid \mathsf{app}\, (\mathsf{lam}\, (\Lambda x.M))\, N \dashrightarrow_\beta (\Lambda x.M)\, N \longrightarrow \mathsf{app}\, (\mathsf{lam}\, (\Lambda x.M))\, N \dashrightarrow_\beta M\{N/x\}}}{\Gamma;\, \Phi_{\mathcal{L}} \mid \Pi N : \mathsf{exp}.\mathsf{app}\, (\mathsf{lam}\, (\Lambda x.M))\, N \dashrightarrow_\beta (\Lambda x.M)\, N \longrightarrow \mathsf{app}\, (\mathsf{lam}\, (\Lambda x.M))\, N \dashrightarrow_\beta M\{N/x\}}}{\Gamma;\, \Phi_{\mathcal{L}} \mid \Pi M : \mathsf{exp} \to \mathsf{exp}.\Pi N : \mathsf{exp}.\mathsf{app}\, (\mathsf{lam}\, M)\, N \dashrightarrow_\beta M\, N \longrightarrow \mathsf{app}\, (\mathsf{lam}\, (\Lambda x.M))\, N \dashrightarrow_\beta M\{N/x\}}}{\Gamma;\, \Phi_{\mathcal{L}} \Longrightarrow \mathsf{app}\, (\mathsf{lam}\, (\Lambda x.M))\, N \dashrightarrow_\beta M\{N/x\}}$$

Fig. 11 Derivations for Lemma 12

Nominal Isabelle's ability to deal with substitution as a total function is crucial for these proofs. If we were dealing with substitution relationally or as a partial function, we might need to reason explicitly about how to push the substitution through each of the 19 nodes. A related issue is that the rules in $\Phi_{\mathcal{L}}$ are defined by fixing specific name values for the bound variables. We have informally glossed over the choice of these names in this article, but in a formalization, to invert the uses of these rules we need to generate fresh variables and rename the formulas. This again forced us to rely heavily on Nominal Isabelle's support for swapping and Isabelle's simplifier.

Finally, although we eventually want to prove that $\bullet; \Phi_{\mathcal{L}} \Longrightarrow \ulcorner t \urcorner \dashrightarrow_\beta \ulcorner u \urcorner$ implies $t \longrightarrow_\beta u$, this is too strong to use as our induction hypothesis. Of course, we need to generalize from the empty context to a context $\ulcorner X \urcorner$. We also need to generalize the conclusion $\ulcorner t \urcorner \dashrightarrow_\beta \ulcorner u \urcorner$ to $M \dashrightarrow_\beta N$ where M, N are λ^{HO} terms that are definitionally equivalent to $\ulcorner t \urcorner$ and $\ulcorner u \urcorner$ respectively. This is in principle no problem, but in practice can lead to extremely large and unwieldy induction hypotheses if we are not careful.

To summarize, we follow this strategy to prove reflection:

1. Introduce an equivalent variant of $\lambda^{\mathrm{HO}}_{\Rightarrow}$ instrumented with height annotations.
2. Prove inversion principles for focused derivations involving each rule of $\Phi_{\mathcal{L}}$.
3. Introduce a convenient definition $X \vdash t \leftrightharpoons M$ with useful properties, particularly that $\ulcorner X \urcorner; \Phi_{\mathcal{L}} \Longrightarrow M \dashrightarrow_\beta N$ implies that for some t, u we have $X \vdash t \leftrightharpoons M$ and $X \vdash u \leftrightharpoons N$ where $t \longrightarrow_\beta u$.
4. Conclude that for closed t, u, if $\bullet; \Phi_{\mathcal{L}} \Longrightarrow \ulcorner t \urcorner \dashrightarrow_\beta \ulcorner u \urcorner$ then $t \longrightarrow_\beta u$.

Height-bounded rules for $\lambda^{\mathrm{HO}}_{\Rightarrow}$ To formalize the informal proof idea sketched above, we introduce an instrumented version of the $\lambda^{\mathrm{HO}}_{\Rightarrow}$ proof rules. We define the two instrumented judgments $\Gamma; \Phi \Longrightarrow_n \phi$ and $\Gamma; \Phi \mid \phi \longrightarrow_n A$ in Fig. 12. Note that n is only incremented in the rule $(\mathrm{sel})^n$, so n essentially tracks the number of uses of that rule.

We will need the following basic properties:

Proposition 5 (Properties of height-bounded derivations)

1. *If* $n \leq m$ *then* $\Gamma; \Phi \Longrightarrow_n \phi$ *implies* $\Gamma; \Phi \Longrightarrow_m \phi$.
2. *If* $n \leq m$ *then* $\Gamma; \Phi \mid \phi \longrightarrow_n A$ *implies* $\Gamma; \Phi \mid \phi \longrightarrow_m A$.
3. $\Gamma; \Phi \Longrightarrow_n \phi$ *holds for some* n *iff* $\Gamma; \Phi \Longrightarrow \phi$.
4. $\Gamma; \Phi \mid \phi \longrightarrow_n A$ *holds for some* n *iff* $\Gamma; \Phi \mid \phi \longrightarrow A$.

$$\frac{\phi \in \Phi \quad \Gamma; \Phi \mid \phi \longrightarrow_n A}{\Gamma; \Phi \Longrightarrow_{n+1} A}\ (\mathrm{sel})^n \qquad \frac{\Gamma \vdash A = A' : o}{\Gamma; \Phi \mid A' \longrightarrow_n A}\ (\mathrm{hyp})^n$$

$$\frac{\Gamma; \Phi, \phi \Longrightarrow_n \psi}{\Gamma; \Phi \Longrightarrow_n \phi \supset \psi}\ (\supset R)^n \qquad \frac{\Gamma; \Phi \Longrightarrow_n \phi \quad \Gamma; \Phi \mid \psi \longrightarrow_n A}{\Gamma; \Phi \mid \phi \supset \psi \longrightarrow_n A}\ (\supset L)^n$$

$$\frac{\Gamma, x{:}\tau; \Phi \Longrightarrow_n \phi}{\Gamma; \Phi \Longrightarrow_n \Pi x{:}\tau.\phi}\ (\Pi R)^n \qquad \frac{\Gamma \vdash M : \tau \quad \Gamma; \Phi \mid \phi\{M/x\} \longrightarrow_n A}{\Gamma; \Phi \mid \Pi x{:}\tau.\phi \longrightarrow_n A}\ (\Pi L)^n$$

Fig. 12 Height-bounded $\lambda^{\mathrm{HO}}_{\Rightarrow}$ derivations

Proof The first two parts are by an easy simultaneous structural induction on derivations. Parts (3) and (4) are easy in the forward direction; in the reverse direction we need the monotonicity properties (parts (1,2)) for the case $(\supset L)$. □

Inversion principles for the reduction rules We will now show that for each of the formulas $\phi \in \{(\text{red\_app}_1), (\text{red\_app}_2), (\text{red\_lam}), (\text{red\_beta})\} = \Phi_{\mathcal{L}}$, if we know $\Gamma; \Phi_{\mathcal{L}} \mid \phi \longrightarrow_n M \dashrightarrow_\beta N$, then the derivation can be inverted in an appropriate way. To show this, we will frequently need to use the fact that the $\dashrightarrow_\beta$ relation symbol is injective with respect to definitional equivalence. This follows easily from properties already established, but it is needed frequently so we state it as a separate lemma:

Lemma 13 (Injectivity of reduction) *If* $\Gamma \vdash (M \dashrightarrow_\beta N) = (M' \dashrightarrow_\beta N') : o$ *then* $\Gamma \vdash M = M' : \mathsf{exp}$ *and* $\Gamma \vdash N = N' : \mathsf{exp}$.

Proof We know that both $\Gamma \vdash M \dashrightarrow_\beta N : o$ and $\Gamma \vdash M' \dashrightarrow_\beta N' : o$ must hold. By the canonical-forms theorem, we know both terms canonicalize to a common canonical form. By inversion and inspection of the derivations leading to the canonical forms, we can see that the canonical form must be of the form $K \dashrightarrow_\beta K'$, and moreover we must have subderivations of $\Gamma \vdash M \Uparrow K : \mathsf{exp}$ and $\Gamma \vdash M' \Uparrow K : \mathsf{exp}$. So by soundness of canonicalization (Theorem 4) and transitivity and symmetry of definitional equivalence we can conclude $\Gamma \vdash M = M' : \mathsf{exp}$. A similar argument shows $\Gamma \vdash N = N' : \mathsf{exp}$, as desired. □

Lemma 14 (Rule reflection)

1. *If* $\Gamma; \Phi_{\mathcal{L}} \mid (\text{red\_app}_1) \longrightarrow_n M \dashrightarrow_\beta N$ *then there exist* M', N', P *such that*
   (a) $\Gamma \vdash M = \mathsf{app}\ M'\ N' : \mathsf{exp}$ *and* $\Gamma \vdash N = \mathsf{app}\ P\ N' : \mathsf{exp}$;
   (b) $\Gamma \vdash M' : \mathsf{exp}$ *and* $\Gamma \vdash N' : \mathsf{exp}$ *and* $\Gamma \vdash P : \mathsf{exp}$; *and*
   (c) $\Gamma; \Phi_{\mathcal{L}} \Longrightarrow_n M' \dashrightarrow_\beta P$.
2. *If* $\Gamma; \Phi_{\mathcal{L}} \mid (\text{red\_app}_2) \longrightarrow_n M \dashrightarrow_\beta N$ *then there exist* M', N', P *such that*
   (a) $\Gamma \vdash M = \mathsf{app}\ M'\ N' : \mathsf{exp}$ *and* $\Gamma \vdash N = \mathsf{app}\ M'\ P : \mathsf{exp}$;
   (b) $\Gamma \vdash M' : \mathsf{exp}$ *and* $\Gamma \vdash N' : \mathsf{exp}$ *and* $\Gamma \vdash P : \mathsf{exp}$; *and*
   (c) $\Gamma; \Phi_{\mathcal{L}} \Longrightarrow_n N' \dashrightarrow_\beta P$.
3. *If* $\Gamma; \Phi_{\mathcal{L}} \mid (\text{red\_lam}) \longrightarrow_n M \dashrightarrow_\beta N$ *then there exist* M', N', x *such that:*
   (a) $x \notin FV(\Gamma, M', N')$;
   (b) $\Gamma \vdash M = \mathsf{lam}\ M' : \mathsf{exp}$ *and* $\Gamma \vdash N = \mathsf{lam}\ N' : \mathsf{exp}$;
   (c) $\Gamma \vdash M' : \mathsf{exp} \to \mathsf{exp}$ *and* $\Gamma \vdash N' : \mathsf{exp} \to \mathsf{exp}$; *and*
   (d) $\Gamma, x{:}\mathsf{exp}; \Phi_{\mathcal{L}} \Longrightarrow_n M'\ x \dashrightarrow_\beta N'\ x$.
4. *If* $\Gamma; \Phi_{\mathcal{L}} \mid (\text{red\_beta}) \longrightarrow_n M \dashrightarrow_\beta N$ *then there exist* M', N', x *such that:*
   (a) $x \notin FV(\Gamma, N')$;
   (b) $\Gamma \vdash M = \mathsf{app}\ (\mathsf{lam}\ (\Lambda x.M'))\ N' : \mathsf{exp}$ *and* $\Gamma \vdash N = (\Lambda x.M')\ N' : \mathsf{exp}$; *and*
   (c) $\Gamma, x{:}\mathsf{exp} \vdash M' : \mathsf{exp}$ *and* $\Gamma \vdash N' : \mathsf{exp}$.

Proof The proofs of these inversion principles require choosing fresh names for the bound variables in the rules and applying case-analysis and inversion rules following the syntax of the rules, using Lemma 13 as needed. Essentially, we show that the derivations can only have the forms exhibited in the proof of Lemma 12. □

To prove the main reflection result, we will prove the more general induction hypothesis:

$$\ulcorner X \urcorner; \Phi_{\mathcal{L}} \Longrightarrow_n M \dashrightarrow_\beta N \text{ implies } \exists t, u.\ X \vdash t \leftrightharpoons M \wedge X \vdash u \leftrightharpoons N \wedge t \longrightarrow_\beta u$$

where $X \vdash t \leftrightharpoons M$ abbreviates the property that M is definitionally equivalent to the encoding of t with respect to variables in X:

$$X \vdash t \leftrightharpoons M \triangleq \ulcorner X \urcorner \vdash \ulcorner t \urcorner = M : \mathsf{exp}$$

It is straightforward to show from prior developments that this relation satisfies the following properties, which are needed in the proof of reflection:

Proposition 6 (Properties of encoding)

1. *(Variable) If* $x \in X$ *then* $X \vdash x \leftrightharpoons x$.
2. *(Application) If* $X \vdash t \leftrightharpoons M$ *and* $X \vdash u \leftrightharpoons N$ *then* $X \vdash t\ u \leftrightharpoons \mathsf{app}\ M\ N$.
3. *(Lambda) If* $X, x \vdash t \leftrightharpoons M$ *and* $x \notin FV(X)$ *then* $X \vdash \lambda x.t \leftrightharpoons \mathsf{lam}\ (\Lambda x.M)$.
4. *(Closure under definitional equivalence) If* $\ulcorner X \urcorner \vdash M = N : \mathsf{exp}$ *then* $X \vdash t \leftrightharpoons M$ *holds if and only if* $X \vdash t \leftrightharpoons N$ *holds.*
5. *(Injectivity) If* $X \vdash t \leftrightharpoons M$ *and* $X \vdash u \leftrightharpoons M$ *then* $t = u$.
6. *(Surjectivity) If* $\ulcorner X \urcorner \vdash M : \mathsf{exp}$ *then there exists* t *with* $X \vdash t \leftrightharpoons M$.
7. *(Substitution) If* $X, x \vdash t \leftrightharpoons M$ *and* $X \vdash u \leftrightharpoons N$ *where* $x \notin FV(X)$ *then* $X \vdash t[u/x] \leftrightharpoons M\{N/x\}$.

Proof Parts 1–3 are immediate by unwinding definitions. Injectivity and surjectivity follow by Theorems 7 and 9. Most other parts follow by combining previously established properties for the individual judgments. For part 5 we need Theorem 7. For part 6 we need Lemma 6 and Theorem 9. Part 7 requires Lemma 6 and Theorem 10. □

We can now establish the main result using the height-instrumented derivations, as follows:

Theorem 12 *If* $\ulcorner X \urcorner; \Phi_{\mathcal{L}} \Longrightarrow_n M \dashrightarrow_\beta N$ *then there exist* t, u *with* $X \vdash t \leftrightharpoons M$ *and* $X \vdash u \leftrightharpoons N$ *and* $t \longrightarrow_\beta u$.

Proof We proceed by induction on n. If $n = 0$, the conclusion holds vacuously by case analysis since $\ulcorner X \urcorner; \Phi_{\mathcal{L}} \Longrightarrow_0 M \dashrightarrow_\beta N$ can never be derived. If $n = n_0 + 1$ where the induction hypothesis holds for n_0, then we distinguish cases. Since the conclusion of the rule is an atomic formula, the derivation must be of the form:

$$\frac{\phi \in \Phi_{\mathcal{L}} \quad \ulcorner X \urcorner; \Phi_{\mathcal{L}} \mid \phi \longrightarrow_{n_0} M \dashrightarrow_\beta N}{\ulcorner X \urcorner; \Phi_{\mathcal{L}} \Longrightarrow_{n_0+1} M \dashrightarrow_\beta N}$$

There are only four choices for ϕ, and for each possibility there is an applicable case of Lemma 14.

1. (red_app$_1$): Using Lemma 14(1), we know that there must exist well-formed M', N', P such that M is definitionally equivalent to $\mathsf{app}\ M'\ N'$ and N is definitionally equivalent to $\mathsf{app}\ P\ N'$ and $\ulcorner X \urcorner; \Phi_{\mathcal{L}} \Longrightarrow_{n_0} M' \dashrightarrow_\beta P$. So, by induction, we know that there must exist t and t' with $X \vdash t \leftrightharpoons M'$ and $X \vdash t' \leftrightharpoons P$ and $t \longrightarrow_\beta t'$. Moreover, by Proposition 6(6), we also know that there is a u such that $X \vdash u \leftrightharpoons N'$. Hence, to conclude, using Proposition 6(2) we can derive:

$$\frac{X \vdash t \leftrightharpoons M' \quad X \vdash u \leftrightharpoons N'}{X \vdash t\ u \leftrightharpoons \mathsf{app}\ M'\ N'} \qquad \frac{X \vdash t' \leftrightharpoons P \quad X \vdash u \leftrightharpoons N'}{X \vdash t'\ u \leftrightharpoons \mathsf{app}\ P\ N'} \qquad \frac{t \longrightarrow_\beta t'}{t\ u \longrightarrow_\beta t'\ u}$$

2. (red_app$_2$): Similar to the first case, using Lemma 14(2).
3. (red_lam): Using Lemma 14(3), we know that there must exist M' and N' and x with M definitionally equivalent to $\mathsf{lam}\ M'$ and N definitionally equivalent to $\mathsf{lam}\ N'$ and $\ulcorner X \urcorner, x{:}\mathsf{exp}; \Phi_{\mathcal{L}} \Longrightarrow_{n_0} M'\ x \dashrightarrow_\beta N'\ x$ derivable. So, by induction, we know that there must exist t, u with $X, x \vdash t \leftrightharpoons M'\ x$ and $X, x \vdash u \leftrightharpoons N'\ x$. By transitivity and η-equivalence we know that $\ulcorner X \urcorner \vdash M = \mathsf{lam}\ (\Lambda x.M'\ x) : \mathsf{exp}$, so we can show that $X \vdash \lambda x.t \leftrightharpoons M$ by reasoning as follows:

$$\frac{\ulcorner X \urcorner \vdash M = \mathsf{lam}\ (\Lambda x.M'\ x) : \mathsf{exp} \quad \dfrac{X, x \vdash t \leftrightharpoons M'\ x}{X \vdash \lambda x.t \leftrightharpoons \mathsf{lam}\ (\Lambda x.M'\ x)}}{X \vdash \lambda x.t \leftrightharpoons M}$$

   Similarly, since $\ulcorner X \urcorner \vdash N = \mathsf{lam}\ (\Lambda x.N'\ x) : \mathsf{exp}$ also holds by transitivity and η-equivalence, we can show that $X \vdash \lambda x.u \leftrightharpoons N$. Finally, we can conclude that the desired object β-reduction step can be performed:

$$\frac{t \longrightarrow_\beta u}{\lambda x.t \longrightarrow_\beta \lambda x.u}$$

4. (red_beta): Using Lemma 14(4), we know that there must exist x, M', N' such that $x \notin FV(\Gamma, N')$ and

   (a) $\Gamma \vdash M = \mathsf{app}\ (\mathsf{lam}\ (\Lambda x.M'))\ N' : \mathsf{exp}$ and $\Gamma \vdash N = (\Lambda x.M')\ N' : \mathsf{exp}$; and
   (b) $\Gamma, x{:}\mathsf{exp} \vdash M' : \mathsf{exp}$ and $\Gamma \vdash N' : \mathsf{exp}$.

   By Proposition 6(6), we know that there must exist t, u with $X, x \vdash t \leftrightharpoons M'$ and $X \vdash u \leftrightharpoons N'$. First, since $\ulcorner X \urcorner \vdash M = \mathsf{app}\ (\mathsf{lam}\ (\Lambda x.M'))\ N' : \mathsf{exp}$, we can show that $X \vdash (\lambda x.t)\ u \leftrightharpoons M$, reasoning as follows:

$$\frac{\ulcorner X \urcorner \vdash M = \mathsf{app}\ (\mathsf{lam}\ (\Lambda x.M'))\ N' : \mathsf{exp} \quad \dfrac{\dfrac{X, x \vdash t \leftrightharpoons M'}{X \vdash (\lambda x.t) \leftrightharpoons \mathsf{lam}\ (\Lambda x.M')} \quad X \vdash u \leftrightharpoons N'}{X \vdash (\lambda x.t)\ u \leftrightharpoons \mathsf{app}\ (\mathsf{lam}\ (\Lambda x.M'))\ N'}}{X \vdash (\lambda x.t)\ u \leftrightharpoons M}$$

   Next, observe that $\ulcorner X \urcorner \vdash N = M'\{N'/x\} : \mathsf{exp}$ must hold by transitivity and β-equivalence, so we can also show that $X \vdash t[u/x] \leftrightharpoons N$, reasoning as follows:

$$\frac{\ulcorner X \urcorner \vdash N = M'\{N'/x\} : \mathsf{exp} \quad \dfrac{X, x \vdash t \leftrightharpoons M' \quad X \vdash u \leftrightharpoons N'}{X \vdash t[u/x] \leftrightharpoons M'\{N'/x\}}}{X \vdash t[u/x] \leftrightharpoons N}$$

Finally, we can conclude that the above terms do reduce in the object-language:

$$\overline{(\lambda x.t)\ u \longrightarrow_\beta t[u/x]}$$

Since these were the only four possibilities, the proof is complete. □

Corollary 5 (Encoding relation reflects reduction) *If* $\ulcorner X \urcorner; \Phi_{\mathcal{L}} \Longrightarrow M \dashrightarrow_\beta N$ *then there exist* t, u *with* $X \vdash t \leftrightharpoons M$ *and* $X \vdash u \leftrightharpoons N$ *and* $t \longrightarrow_\beta u$.

Proof By the previous theorem and the equivalence between the height-instrumented and ordinary $\lambda^{\mathrm{HO}}_{\Rightarrow}$ rules. □

Theorem 13 (Encoding reflects reduction) *If* $\ulcorner X \urcorner; \Phi_{\mathcal{L}} \Longrightarrow \ulcorner t \urcorner \dashrightarrow_\beta \ulcorner u \urcorner$ *where* $X \vdash t, u$ *then* $t \longrightarrow_\beta u$.

Proof As shown above, we know that there exist t', u' with $X \vdash t' \leftrightharpoons \ulcorner t \urcorner$ and $X \vdash u' \leftrightharpoons \ulcorner u \urcorner$ and $t' \longrightarrow_\beta u'$. Moreover, clearly $X \vdash t \leftrightharpoons \ulcorner t \urcorner$ and $X \vdash u \leftrightharpoons \ulcorner u \urcorner$. Hence, by injectivity of the encoding relation we have $t = t' \longrightarrow_\beta u' = u$. □

In particular we have established the reflection direction of part 4(c) of the definition of interpretation:

Corollary 6 *If* t, u *are closed terms and* $\bullet; \Phi_{\mathcal{L}} \Longrightarrow \ulcorner t \urcorner \dashrightarrow_\beta \ulcorner u \urcorner$ *then* $t \longrightarrow_\beta u$.

6 Conclusion

We have shown that the model-theoretic notion of interpretation provides a useful generalization of the notion of adequacy in higher-order abstract syntax, and proved the existence of an interpretation in detail for a standard example. We have intentionally focused on a simply-typed (and simple) metalanguage and simple object language in order to avoid an overwhelming number of details. We believe that the resulting formalization will be useful for further studying adequacy in a number of contexts.

An immediate next step for future work is to adapt our formalization to study adequacy for more complex examples, such as Abel's encoding of the $\lambda\mu$-calculus [1]. We are also interested in formalizing adequacy for other systems, particularly for type theories supporting nominal abstract syntax [2, 19]. Our proof may be useful for identifying proof patterns for automating adequacy proofs, or avenues for further improving reasoning about names and binding in Nominal Isabelle.

Most higher-order abstract syntax techniques have the capability to use implication subgoals (viewing theories as higher-order logic programs). For example, a typing rule for λ-abstraction is typically implemented using a rule such as:

$$\Pi T_1, T_2{:}\mathsf{ty}.\Pi M{:}\mathsf{exp} \to \mathsf{exp}.\ (\Pi x{:}\mathsf{exp}.\ \mathsf{of}\ x\ T_1 \to \mathsf{of}\ (M\ x)\ T_2)$$
$$\to \mathsf{of}\ (\mathsf{lam}\ M)\ (\mathsf{arrow}\ T_1\ T_2)$$

where ty is a new type with constructor arrow : ty → ty → ty and of : exp → ty → o. Our proof did not exercise this capability. (Put another way, nowhere did we make use of the ($\supset R$) rule.) Moreover, we have not discussed how to translate object-language properties to metatheoretic properties in Twelf, which rely on additional features such as *worlds* (sets of contexts) and *subordination* relations (independence constraints among types). Formalizing adequacy for richer specifications in languages such as LF or Canonical LF therefore remains a significant open problem. One natural next step could be to formalize Harper and Licata's detailed, but not mechanically checked, development of adequacy [11] for the simply-typed lambda-calculus in LF.

A final question is the (informal) adequacy of our encodings of the object language, meta-language and meta-logic in Nominal Isabelle. Formally, Urban showed that a vanilla untyped λ-calculus is isomorphic to the nominal datatype we used here; a similar result could easily be shown for our λ^{HO} language. Beyond that, as we argued in the introduction, it appears impossible to fully formalize adequacy; instead, all we can formalize is the relationships between two different representations in some third system in which we must repose trust. Our work nevertheless shows that formalization can help increase confidence in the adequacy of representations of languages with name-binding.

Acknowledgements We wish to thank Bob Harper and Frank Pfenning for discussions about adequacy and Andrew Pitts and Christian Urban for discussions about nominal techniques and Nominal Isabelle. Cheney is supported by a Royal Society University Research Fellowship. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

Appendix: Proof of Completeness of Canonicalization

The logical relation is shown in Fig. 13. As with other Kripke-style logical relations, we index by a context Γ and quantify over argument terms M', N' over an extended context Γ′ in the case for function application. Note that we build the fact that logically related λ^{HO} terms of base type have a common canonical form into the base case. We also extend the logical relation to handle simultaneous substitutions.

$$\Gamma \vdash M = N \in [\![a]\!] \equiv \exists K.\, \Gamma \vdash M \Uparrow K : a \wedge \Gamma \vdash N \Uparrow K : a$$

$$\Gamma \vdash M = N \in [\![\tau_1 \to \tau_2]\!] \equiv \forall \Gamma' \supseteq \Gamma.\, \forall M', N'.\, \Gamma' \vdash M' = N' \in [\![\tau_1]\!] \Rightarrow \Gamma' \vdash M\ M' = N\ N' \in [\![\tau_2]\!]$$

$$\Gamma \vdash \bullet = \bullet \in [\![\bullet]\!] \equiv \top$$

$$\Gamma \vdash \theta, M/x = \sigma, N/x \in [\![\Delta, x{:}\tau]\!] \equiv \Gamma \vdash \theta = \sigma \in [\![\Delta]\!] \wedge \Gamma \vdash M = N \in [\![\tau]\!]$$

Fig. 13 Logical relation for existence of canonical forms

Simultaneous substitutions are of the form $\theta = M_1/x_1, \ldots, M_n/x_n$ and their behavior is defined in the standard way:

$$\begin{aligned} x\{\theta\} &= M \qquad (M/x \in \theta) \\ y\{\theta\} &= y \qquad (\text{no } M/y \in \theta) \\ (M\ N)\{\theta\} &= M\{\theta\}\ N\{\theta\} \\ (\lambda x.M)\{\theta\} &= \lambda x.(M\{\theta\}) \quad (x \notin FV(\theta)) \end{aligned}$$

where $FV(\theta)$ includes all variables mentioned in θ.

The main difference between our proof technique and the other approaches mentioned above is that we directly establish the existence of a common canonical form using the canonicalization judgment. In contrast, the main objective in the earlier proofs is completeness of type-directed $\beta\eta$-equivalence algorithms [5, 12]. Harper and Pfenning [12] showed that canonical and atomic forms can be extracted from their algorithmic equivalence derivations once these have been shown to exist. For our purposes, this approach would represent extra formalization effort that is not strictly necessary for the adequacy results we wish to establish.

Our proof is also technically slightly simpler. Compared to Crary's proof, we use simpler rules that perform weak head reduction one step at a time rather than multi-step weak head normalization. Compared to Harper and Pfenning's proof, we can prove transitivity of the logical relation directly; in contrast, in our formalization of Harper and Pfenning's proof, proving transitivity of the logical relation requires establishing that the equivalence algorithm is transitive, which in turn requires simultaneous induction on two derivations. Here, in contrast, transitivity of the logical relation follows from determinacy, which we proved using structural induction.

Lemma 15 (Logical relation weakening) *If* $\Gamma \subseteq \Gamma'$ *then:*

1. *If* $\Gamma \vdash M = N \in [\![\tau]\!]$ *then* $\Gamma' \vdash M = N \in [\![\tau]\!]$.
2. *If* $\Gamma \vdash \theta = \sigma \in [\![\Delta]\!]$ *then* $\Gamma' \vdash \theta = \sigma \in [\![\Delta]\!]$.

Proof Straightforward by induction on τ or Δ respectively, using weakening for canonicalization in the base case. □

Theorem 14 (Main theorem)

1. *If* $\Gamma \vdash M = N \in [\![\tau]\!]$ *then there exists* K *such that* $\Gamma \vdash M \Uparrow K : \tau$ *and* $\Gamma \vdash N \Uparrow K : \tau$.
2. *If* $\Gamma \vdash M \downarrow K : \tau$ *and* $\Gamma \vdash N \downarrow K : \tau$ *hold for some* K *then* $\Gamma \vdash M = N \in [\![\tau]\!]$.

Proof We prove both implications simultaneously (for arbitrary Γ, M, N) by induction on τ. The base cases are trivial. If $\tau = \tau_1 \to \tau_2$, where the induction hypothesis holds for τ_1 and τ_2, then we proceed as follows.

1. For part (1), let Γ, M, N be given such that $\Gamma \vdash M = N \in [\![\tau_1 \to \tau_2]\!]$. Choose a fresh variable name $x \notin FV(\Gamma, M, N)$. Then $\Gamma, x{:}\tau_1$ is a well-formed context extending Γ, and we can show easily that $\Gamma, x{:}\tau_1 \vdash x \downarrow x : \tau_1$. Hence, by induction we know $\Gamma, x{:}\tau_1 \vdash x = x \in [\![\tau_1]\!]$. By definition of the logical relation, it follows that $\Gamma, x{:}\tau_1 \vdash M\ x = N\ x \in [\![\tau_2]\!]$. By induction, we can choose K satisfying

$\Gamma, x{:}\tau_1 \vdash M\ x \Uparrow K : \tau_2$ and $\Gamma, x{:}\tau_1 \vdash N\ x \Uparrow K : \tau_2$. Hence, we can conclude that $\Gamma \vdash M \Uparrow \Lambda x.K : \tau_1 \to \tau_2$ and $\Gamma \vdash N \Uparrow \Lambda x.K : \tau_1 \to \tau_2$, i.e., $\Lambda x.K$ is the common canonical form of M and N.

2. For part (2), if $\tau = \tau_1 \to \tau_2$, let Γ, M, N, K be given such that $\Gamma \vdash M \downarrow K : \tau_1 \to \tau_2$ and $\Gamma \vdash N \downarrow K : \tau_1 \to \tau_2$ hold. Then by definition of the logical relation, it suffices to show that for any $\Gamma' \supseteq \Gamma$ and M', N' satisfying $\Gamma' \vdash M' = N' \in [\![\tau_1]\!]$, we have $\Gamma' \vdash M\ M' = N\ N' \in [\![\tau_2]\!]$. Let Γ', M', N' satisfying these criteria be given. By induction, since $\Gamma' \vdash M' = N' \in [\![\tau_1]\!]$ we also know that there must exist a K' such that $\Gamma \vdash M' \Uparrow K' : \tau_1$ and $\Gamma \vdash N' \Uparrow K' : \tau_1$. Thus, we can derive:

$$\dfrac{\dfrac{\Gamma \vdash M \downarrow K : \tau_1 \to \tau_2}{\Gamma' \vdash M \downarrow K : \tau_1 \to \tau_2} \quad \Gamma' \vdash M' \Uparrow K' : \tau_1}{\Gamma' \vdash M\ M' \downarrow K\ K' : \tau_2}$$

using weakening and the ($\downarrow_{\text{app}}$) rule. Similarly, we can derive $\Gamma' \vdash N\ N' \downarrow K\ K' : \tau_2$. By induction, we therefore can conclude that $\Gamma' \vdash M\ M' = N\ N' \in [\![\tau_2]\!]$, and since Γ', M', N' were arbitrary we can conclude that $\Gamma \vdash M = N \in [\![\tau_1 \to \tau_2]\!]$ also. □

Lemma 16 (Logical relation symmetry)

1. *If* $\Gamma \vdash M = N \in [\![\tau]\!]$ *then* $\Gamma \vdash N = M \in [\![\tau]\!]$.
2. *If* $\Gamma \vdash \theta = \sigma \in [\![\Delta]\!]$ *then* $\Gamma' \vdash \sigma = \theta \in [\![\Delta]\!]$

Proof Straightforward inductions on τ or Δ. □

Lemma 17 (Logical relation transitivity)

1. *If* $\Gamma \vdash M = N \in [\![\tau]\!]$ *and* $\Gamma \vdash N = N' \in [\![\tau]\!]$ *then* $\Gamma \vdash M = N' \in [\![\tau]\!]$.
2. *If* $\Gamma \vdash \theta = \sigma \in [\![\Delta]\!]$ *and* $\Gamma \vdash \sigma = \sigma' \in [\![\Delta]\!]$ *then* $\Gamma' \vdash \theta = \sigma' \in [\![\Delta]\!]$.

Proof Part (1) is by induction on τ.

- If $\tau = a$ then by assumption we have $\Gamma \vdash M \Uparrow K : a$ and $\Gamma \vdash N \Uparrow K : a$ and $\Gamma \vdash N \Uparrow K' : a$ and $\Gamma \vdash N' \Uparrow K' : a$. By determinacy, we know that $K = K'$. Hence, $\Gamma \vdash M \Uparrow K : a$ and $\Gamma \vdash N' \Uparrow K : a$, so we can conclude $\Gamma \vdash M = N' \in [\![a]\!]$.
- If $\tau = \tau_1 \to \tau_2$ where the induction hypothesis holds for τ_1 and τ_2 then by assumption we have $\Gamma \vdash M = N \in [\![\tau_1 \to \tau_2]\!]$ and $\Gamma \vdash N = N' \in [\![\tau_1 \to \tau_2]\!]$. To prove $\Gamma \vdash M = N' \in [\![\tau_1 \to \tau_2]\!]$, let $\Gamma' \supseteq \Gamma$, M_1, N_1 be given with $\Gamma' \vdash M_1 = N_1 \in [\![\tau_1]\!]$. By symmetry, we know $\Gamma' \vdash N_1 = M_1 \in [\![\tau_1]\!]$ so by induction we know $\Gamma' \vdash M_1 = M_1 \in [\![\tau_1]\!]$. Hence, by assumption we know that $\Gamma' \vdash M\ M_1 = N\ M_1 \in [\![\tau_2]\!]$ and $\Gamma' \vdash M\ M_1 = N'\ N_1 \in [\![\tau_2]\!]$, so by induction we can conclude that $\Gamma \vdash M = N' \in [\![\tau_1 \to \tau_2]\!]$.

For part (2), we proceed by straightforward induction on Δ. □

Lemma 18 (Closure under weak head expansion)

1. *If* $M \xrightarrow{\text{whr}} M'$ *and* $\Gamma \vdash M' = N \in [\![\tau]\!]$ *then* $\Gamma \vdash M = N \in [\![\tau]\!]$
2. *If* $N \xrightarrow{\text{whr}} N'$ *and* $\Gamma \vdash M = N' \in [\![\tau]\!]$ *then* $\Gamma \vdash M = N \in [\![\tau]\!]$

Proof By (straightforward) induction on τ, using ($\Uparrow_{\text{whr}}$) in the base case. □

Theorem 15 (Well-formed terms are logically related) *If* $\Gamma \vdash M : \tau$ *then for any* Δ, θ, σ, *if* $\Delta \vdash \theta = \sigma \in [\![\Gamma]\!]$ *then* $\Delta \vdash M\{\theta\} = M\{\sigma\} \in [\![\tau]\!]$.

Proof By induction on the well-formedness derivation. For variables and applications the reasoning is straightforward. For constants, we appeal to Theorem 14. For Λ-abstractions, suppose we have

$$\frac{\Gamma, x{:}\tau_1 \vdash M : \tau_2}{\Gamma \vdash \Lambda x.M : \tau_1 \to \tau_2}$$

and assume that for some Δ, θ, σ we have $\Delta \vdash \theta = \sigma \in [\![\Gamma]\!]$. We wish to show $\Delta \vdash M\{\theta\} = M\{\sigma\} \in [\![\tau_1 \to \tau_2]\!]$, so let $\Delta' \supseteq \Delta$, M', N' be given with $\Delta' \vdash M' = N' \in [\![\tau_1]\!]$. Then using weakening and the definition of the logical relation for substitutions we have $\Delta' \vdash \theta, M'/x = \sigma, N'/x \in [\![\Gamma, x{:}\tau]\!]$. It follows by induction that $\Delta' \vdash M\{\theta, M'/x\} = M\{\sigma, N'/x\} \in [\![\tau_2]\!]$. By closure under weak head expansion and properties of simultaneous substitutions, we have $\Delta' \vdash (\Lambda x.M\{\theta\})\ M' = (\Lambda x.M\{\sigma\})\ N' \in [\![\tau_2]\!]$. Since Γ', M', N' were arbitrary, we can conclude that $\Delta \vdash (\Lambda x.M\{\theta\}) = (\Lambda x.M\{\sigma\}) \in [\![\tau_1 \to \tau_2]\!]$, as desired. □

Lemma 19 (Closure under $\beta\eta$-conversion)

1. *If* $\Gamma \vdash (\Lambda x.M)\ N : \tau$ *and* $\Gamma' \vdash \theta = \sigma \in [\![\Gamma]\!]$ *and* $x \notin FV(\Gamma, \Gamma', \theta, \sigma, N)$ *then* $\Gamma' \vdash (\Lambda x.M\{\theta\})\ N\{\theta\} = M\{N/x\}\{\sigma\} \in [\![\tau]\!]$.
2. *If* $\Gamma \vdash M : \tau \to \tau'$ *and* $\Gamma' \vdash \theta = \sigma \in [\![\Gamma]\!]$ *and* $x \notin FV(\Gamma, \Gamma', \theta, \sigma, M)$ *then* $\Gamma' \vdash M\{\theta\} = \Lambda x.M\{\sigma\}\ x \in [\![\tau \to \tau']\!]$.

Proof Both parts are straightforward, using subject reduction, the previous theorem, and closure under weak head expansion. □

Theorem 16 (Definitionally equivalent terms are logically related) *If* $\Gamma \vdash M = N : \tau$ *and* $\Gamma' \vdash \theta = \sigma \in [\![\Gamma]\!]$ *then* $\Gamma' \vdash M\{\theta\} = N\{\sigma\} \in [\![\tau]\!]$.

Proof The proof for the cases of congruence rules are similar to those for Theorem 15. The new cases involve β- or η-equivalence, symmetry and transitivity. The cases for β- and η-equivalence follow from the previous lemma, and the cases for symmetry and transitivity follow using symmetry and transitivity of the logical relation. □

Lemma 20 (Identity substitution is logically related to itself) *For any* Γ *we have* $\Gamma \vdash \mathsf{id}_\Gamma = \mathsf{id}_\Gamma \in [\![\Gamma]\!]$.

Lemma 21 (Definitionally equivalent implies logically related) *If* $\Gamma \vdash M = N : \tau$ *then* $\Gamma \vdash M = N \in [\![\tau]\!]$.

Proof Immediate, chaining Lemma 20 and Theorem 16. □

We can now complete the proof of Theorem 6 as follows:

Proof (of Theorem 6) By the previous lemma we have $\Gamma \vdash M = N \in [\![\tau]\!]$, and by the main theorem (Theorem 14) we have $\Gamma \vdash M \Uparrow K : \tau$ and $\Gamma \vdash N \Uparrow K : \tau$ for some K. Clearly, by soundness (Theorem 4) K is a canonical form definitionally equivalent to both M and N and by determinacy (Theorem 5) it is unique. □

References

1. Abel, A.: A third-order representation of the $\lambda\mu$-calculus. In: Ambler, S., Crole, R., Momigliano, A. (eds.) Electronic Notes in Theoretical Computer Science, vol. 58. Elsevier Science Publishers (2001)
2. Cheney, J.: A simple nominal type theory. In: Logical Frameworks and Meta-Languages: Theory and Practice, pp. 90–104 (2008)
3. Cheney, J., Urban, C.: Nominal logic programming. ACM Trans. Program. Lang. Syst. **30**(5), 26 (2008)
4. Coquand, T.: An algorithm for testing conversion in type theory. In: Logical Frameworks, pp. 255–279. Cambridge University Press, New York (1991)
5. Crary, K.: Logical relations and a case study in equivalence checking. In: Pierce, B.C. (ed.) Advanced Topics in Types and Programming Languages, pp. 223–244. MIT Press (2005)
6. Crary, K., Harper, R.: Higher-order abstract syntax: setting the record straight. SIGACT News **37**(3), 93–96 (2006)
7. Crole, R.: The representational adequacy of hybrid. Math. Struct. Comput. Sci. (2011, to appear)
8. Gabbay, M.J., Pitts, A.M.: A new approach to abstract syntax with variable binding. Form. Asp. Comput. **13**, 341–363 (2002)
9. Gardner, P.: Equivalences between logics and their representing type theories. Math. Struct. Comput. Sci. **5**(3), 323–349 (1995)
10. Harper, R., Honsell, F., Plotkin, G.D.: A framework for defining logics. J. ACM **40**(1), 143–184 (1993)
11. Harper, R., Licata, D.R.: Mechanizing metatheory in a logical framework. J. Funct. Program. **17**(4–5), 613–673 (2007)
12. Harper, R., Pfenning, F.: On equivalence and canonical forms in the LF type theory. ACM Trans. Comput. Log. **6**(1), 61–101 (2005)
13. Hodges, W.: A Shorter Model Theory. Cambridge University Press (1997)
14. Miller, D., Nadathur, G.: A logic programming approach to manipulating formulas and programs. In: Haridi, S. (ed.) IEEE Symposium on Logic Programming, pp. 379–388. San Francisco (1987)
15. Miller, D., Nadathur, G., Pfenning, F., Scedrov, A.: Uniform proofs as a foundation for logic programming. Ann. Pure Appl. Logic **51**, 125–157 (1991)
16. Narboux, J., Urban, C.: Formalising in nominal Isabelle Crary's completeness proof for equivalence checking. In: LFMTP, vol. 196 of ENTCS (2007)
17. Norrish, M., Vestergaard, R.: Proof pearl: De Bruijn terms really do work. In: Schneider, K., Brandt, J. (eds.) TPHOLs. Lecture Notes in Computer Science, vol. 4732, pp. 207–222. Springer (2007)
18. Pfenning, F.: Logical frameworks. In: Robinson, A., Voronkov, A. (eds.) Handbook of Automated Reasoning, vol. II, chapter 17, pp. 1063–1147. Elsevier Science (2001)
19. Pitts, A.: Nominal system T. In: POPL, pp. 159–170 (2010)
20. Pitts, A.M.: Alpha-structural recursion and induction. J. ACM **53**(3), 459–506 (2006)
21. Urban, C.: Nominal techniques in Isabelle/HOL. J. Autom. Reason. **40**(4), 327–356 (2008)
22. Urban, C., Berghofer, S., Norrish, M.: Barendregt's variable convention in rule inductions. In: CADE. LNAI, vol. 4603, pp. 35–50 (2007)
23. Urban, C., Cheney, J., Berghofer, S.: Mechanizing the metatheory of LF. ACM Trans. Comput. Log. (2010, in press)
24. Watkins, K., Cervesato, I., Pfenning, F., Walker, D.: A concurrent logical framework I: judgments and properties. Technical Report CMU-CS-02-101, Carnegie Mellon University (2003)