Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 375 (2007) 201–226
Theoretical Computer Science
www.elsevier.com/locate/tcs

The Girard–Reynolds isomorphism (second edition)

Philip Wadler

Department of Informatics, University of Edinburgh, James Clerk Maxwell Building, The King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, Scotland, United Kingdom

Abstract

Jean-Yves Girard and John Reynolds independently discovered the second-order polymorphic lambda calculus, F2. Girard additionally proved a *Representation Theorem*: every function on natural numbers that can be proved total in second-order intuitionistic predicate logic, P2, can be represented in F2. Reynolds additionally proved an *Abstraction Theorem*: every term in F2 satisfies a suitable notion of logical relation; and formulated a notion of *parametricity* satisfied by well-behaved models.

We observe that the essence of Girard's result is a projection from P2 into F2, and that the essence of Reynolds's result is an embedding of F2 into P2, and that the Reynolds embedding followed by the Girard projection is the identity. We show that the inductive naturals are exactly those values of type natural that satisfy Reynolds's notion of parametricity, and as a consequence characterize situations in which the Girard projection followed by the Reynolds embedding is also the identity.

An earlier version of this paper used a logic over untyped terms. This version uses a logic over typed term, similar to ones considered by Abadi and Plotkin and by Takeuti, which better clarifies the relationship between F2 and P2.

This paper uses colour to enhance its presentation. If the link below is not blue, follow it for the colour version.

http://homepages.inf.ed.ac.uk/wadler

Keywords: Girard–Reynolds type system; System F; Polymorphic lambda calculus; Abstraction Theorem; Representation Theorem; Curry–Howard

1. Introduction

Double-barrelled names in science indicate cooperation or coincidence: they may label ideas refined by two collaborators or ideas revealed by two discoverers. *Curry–Howard* is a name of the first sort that guarantees the existence of names of the second sort, such as *Hindley–Milner* and *Girard–Reynolds*.

The Curry–Howard isomorphism consists of a correspondence between logic and computation. Propositions correspond to types and proofs correspond to terms. Further, reduction of proofs corresponds to reduction of terms, hence we have no mere bijection but a true isomorphism.

Curry formulated this principle for combinatory logic and combinator terms [6], and Howard observed that it applies to intuitionistic logic and typed lambda calculus [18]. The correspondence extends to a logic with propositional variables and a calculus with type variables, which explains why the logician Roger Hindley and the computer scientist Robin Milner independently discovered the type system underlying ML and Haskell [17,26,10]. It also

E-mail address: wadler@inf.ed.ac.uk.

0304-3975/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2006.12.042

extends to a logic with quantifiers over propositions and a calculus with quantifiers over types, which explains why the logician Jean-Yves Girard and the computer scientist John Reynolds independently discovered the polymorphic lambda calculus [14,35,36].

Girard and Reynolds each made additional discoveries about the calculus that bears their name, here referred to as F2. Girard proved a *Representation Theorem*: every function on natural numbers that can be proved total in second-order predicate calculus P2 can be represented in F2. Reynolds proved an *Abstraction Theorem*: every term in F2 satisfies a suitable notion of logical relation; and formulated a notion of *parametricity* satisfied by well-behaved models. (The presentation of P2 in this paper is similar in spirit but different in detail to that used by Girard; in particular, the version used here is typed, as discussed below.)

The calculus P2 is larger than the image under the Curry–Howard isomorphism of F2: the former has three quantifiers (over individuals, types, and predicates), while the latter has only one (over types). Nonetheless, the essence of Girard's result is a projection from P2 onto F2 that is similar to the Curry–Howard isomorphism, in that it takes propositions to types and proofs to terms, but differs in that it erases information about individuals. This mapping preserves reductions, hence it is no mere surjection but a true epimorphism.

Reynolds's result traditionally concerns binary relations, but it extends to other notions of relation, including a degenerate unary case. In the unary version, the essence of Reynolds's result is an embedding from F2 into P2 that is similar to the Curry–Howard isomorphism, in that it takes types to predicates and terms to proofs, but differs in that it adds information about individuals. This mapping too preserves reductions, hence it is no mere injection but a true monomorphism.

Furthermore, the result on binary relations can be recovered from the result on unary relations by a doubling operation, an embedding from P2 into P2 that takes formulas into formulas, proofs into proofs, and preserves reductions. Reynolds's Abstraction Theorem is an immediate consequence of the Reynolds embedding followed by doubling.

Christopher Strachey distinguished two types of polymorphism, where the meaning of a term depends upon a type [43]. In *parametric* polymorphism, the meaning of the term varies uniformly with the type (an example is the length function), while in *ad hoc* polymorphism, the meaning of the term at different types may not be related (an example is plus, which may have quite different meanings on integers, floats, and strings). Reynolds introduced a *parametricity* condition to capture a semantic notion corresponding to Strachey's parametric polymorphism.

The Reynolds embedding followed by the Girard projection is the identity. Remarkably, I can find no place in the literature where this is remarked! While reading between the lines suggests that some researchers have intuitively grasped that there is a connection between the constructs underlying Reynolds's and Girard's proofs, its precise description seems to have been more elusive.

Going the other way, it is unreasonable to expect that the Girard projection followed by the Reynolds embedding should also yield the identity, because the projection discards all information about individuals.

We will demonstrate the above by considering two different approaches to defining the naturals. Here is the standard inductive definition of the naturals in P2.

$$\mathbf{N} \equiv \{n^{\mathsf{N}} \mid \forall \mathcal{X}^{\mathsf{N}}.\, (\forall m^{\mathsf{N}}.\, m \in \mathcal{X} \to \mathsf{s}\, m \in \mathcal{X}) \to \mathsf{z} \in \mathcal{X} \to n \in \mathcal{X}\}$$

Here N is the type of naturals (an uninterpreted sort), and s denotes successor (an uninterpreted term of sort $\mathsf{N} \to \mathsf{N}$) and z denotes zero (an uninterpreted term of sort N). The above statement gives the usual induction principle for the naturals—if a property $\mathcal{X}$ holds for zero, and if whenever property $\mathcal{X}$ holds for m then it also holds for the successor of m, then the property holds for every natural number n—and then defines $\mathbf{N}$ to be the predicate that holds exactly for such n. It is not hard to prove the following.

$$n \in \mathbf{N} \to \mathsf{s}\, n \in \mathbf{N} \qquad\qquad \mathsf{z} \in \mathbf{N}$$

(The formal proofs appear in Figs. 6 and 7.) Hence, the terms satisfying $\mathbf{N}$ must include z, $\mathsf{s}\,\mathsf{z}$, $\mathsf{s}\,(\mathsf{s}\,\mathsf{z})$, and so on. We name terms satisfying this predicate the *inductive* naturals.

By definition, a term can only satisfy predicate $\mathbf{N}$ if it has type N, but we make no assumption in the opposite direction. In some models there may be terms of type N that do not satisfy $\mathbf{N}$. For example, in a model that supports fixpoints at all types, bottom would be such a term.

So far, we have taken N, s, and z to be uninterpreted: the definition of $\mathbf{N}$ makes sense for any type N whatsoever, and for any terms s and z whatsoever, so long as they have the given types.

Now, however, we apply the Girard projection to derive particular definitions of these. In particular, applying the Girard projection to the predicate **N** in P2 yields a type in F2, which we take to be N.

$$\mathsf{N} \equiv \mathbf{N}^\circ \equiv \forall X. (X \to X) \to (X \to X)$$

And applying the Girard projection to the two proofs of the statements in P2 above yields two terms in F2, which we take to be z and s. (The precise terms appear in Figs. 6 and 7.) The type derived for N is the usual type of the Church numerals, and the terms derived for z and s are the usual definitions of zero and successor on the Church numerals.

Having used the Girard projection to go from P2 to F2, we now use the Reynolds embedding to go back the other way. Applying the Reynolds embedding to the type s in F2 yields the following predicate in P2.

$$\mathsf{N}^* \equiv \{n^{\mathsf{N}} \mid \forall X. \boldsymbol{\forall \mathcal{X}^X.} \forall s^{X\to X}. (\forall m^X. m \in \boldsymbol{\mathcal{X}} \boldsymbol{\to} s\ m \in \boldsymbol{\mathcal{X}}) \boldsymbol{\to} \forall z^X. z \in \boldsymbol{\mathcal{X}} \boldsymbol{\to} n\ X\ s\ z \in \boldsymbol{\mathcal{X}}\}$$

This predicate does not look much like **N**. Unlike before, it makes sense only if N is the type of the Church numerals, since it applies n to specific arguments; and it does not mention s or z. We name terms satisfying this predicate the *deductive* naturals.

In short, we can use arbitrary definitions of N, s, and z, for the inductive naturals, but we must fix these definitions to correspond to the Church numerals in order to define the deductive naturals. Having done so, it is easy to show that every inductive natural is also a deductive natural. The converse does not hold in general. However, we will show the inductive naturals are exactly the deductive naturals that satisfy Reynolds's notion of binary parametricity.

Thus, in the important case where binary parametricity holds for the Church numerals, not only does the Girard projection take **N** to N, but also the Reynolds embedding takes N to **N**, and we have no mere embedding–projection pair but a true isomorphism.

Given a proof $\boldsymbol{m}$ that a term n is an inductive natural, the Girard projection yields a lambda term $\boldsymbol{m}^\circ$, and then the Reynolds embedding yields a proof that this new term is a deductive natural. Using a realizability technique due to Krivine and Parigot [20], we will further show that $\boldsymbol{m}^\circ = n$, strengthening the sense in which the Girard projection and the Reynolds embedding are inverses. From this result we will derive Girard's Representation Theorem.

Girard attributes the representation of the natural numbers in polymorphic lambda calculus to Per Martin-Löf [25] (the original paper was written in 1970, but only published recently). The natural numbers are a special case of an algebraic type. The representation of algebraic types in polymorphic lambda calculus was first proposed by Böhm and Berarducci [4], who characterized the algebraic types as equivalent to polymorphic types of rank two with all qualifiers on the outside. A closely related treatment of algebraic types as *data systems* has been explored by Leivant [21,22] and Krivine and Parigot [20]. All of the results given here for naturals extend straightforwardly to any data system.

Proofs of Reynolds's Abstraction Theorem and Girard's Representation Theorem will emerge naturally from our development, almost as corollaries. These are not so much new proofs, as old proofs clarified. Setting Girard's and Reynolds's proofs in a common framework highlights the relationship between them.

1.1. Relation to other work

Both Girard's and Reynolds's results have spawned large bodies of related work. Girard's Representation Theorem has been further explored by Leivant [21,22] and by Krivine and Parigot [20], among others. Reynolds's Abstraction Theorem and parametricity has been further explored by Reynolds [37,38,29], Reynolds and Plotkin [40], Pitts [30–32], Bainbridge, Freyd, Scedrov and Scott [3], Robinson and Rosolini [41], Hasegawa [16], and Wadler [45,46], among others. Formulations of the Abstraction Theorem and parametricity in terms of logics have been examined by Mairson [23], in various combinations by Abadi, Cardelli, Curien, and Plotkin [1,33,34], and by Takeuti [44]. Tutorials have been written by Girard, Taylor, and Lafont [15] and Reynolds [39].

Moggi [27], Breazu-Tannen and Coquand [5], Freyd [12] and Hyland, Robinson, and Rosolini [19] give models where the type of Boolean has just two elements or the type of N contains just the Church numerals. Ivar Rummelhoff [42] shows that there are PER models where the interpretation of N contains a "nonstandard" value that is not a Church numeral, and hence does not satisfy binary parametricity. Rasmus Møgelberg [28] claims that in every PER model every value of N does satisfy unary parametricity. This shows that unary parametricity does not imply binary parametricity.

In addition to the work of Girard and Reynolds, particularly strong influences on this work include: Böhm and Berarducci [4], who first showed how to represent algebraic types in polymorphic lambda calculus; Leivant [22], who presents Girard's result as a projection from a logic into F2; Mairson [23], who presents Reynolds's result as an embedding form F2 into a logic; Krivine and Parigot [20], who present a realizability result similar to the one given here; and Plotkin and Abadi [33] and Takeuti [44], who relate parametricity to a logic of typed terms essentially the same as P2.

The basic structure of the proofs in Section 5 was suggested, independently, by Wadler [46] and Hasegawa [16]. Wadler's proof was not published, but it circulated informally, and influenced the work of Abadi, Cardelli, and Curien [1] and the subsequent work of Plotkin and Abadi [33]. The notion that parametricity implies that algebraic types have the usual universal properties goes back to Reynolds [37], while the converse seems to have first been suggested by Hasegawa [16]. This paper is the first (so far as I know) to observe that the inductive naturals are exactly those values of type N that satisfy parametricity, even when not all values of that type are parametric.

Mairson [23] appears to have grasped the inverse relation between the Reynolds embedding and the Girard projection, though he does not quite manage to state it. However, Mairson does seem to have missed the power of parametricity. He mislabels as "parametricity" the analogue of Reynolds's Abstraction Theorem, and he never states an analogue of Reynolds's parametricity condition or the Identity Extension Lemma. Thus when he writes "proofs of these equivalences still seem to require structural induction, as well as stronger assumptions than parametricity" [23], I believe this is misleading: the equivalences he refers to cannot be proved using the Abstraction Theorem alone, but can indeed be proved in the presence of parametricity.

The Curry–Howard isomorphism has informed the development of powerful lambda calculi with dependent types, such as de Bruijn's Automath [9], Howard's constructions [18], Martin-Löf's type theory [24], Constable's Nuprl [8], Coquand and Huet's calculus of constructions [7], and Barendregt's lambda cube [2]. Each of these calculi introduces dependent types to map quantifiers over individuals into the type system. In contrast, the Girard projection discards all information about individuals. To quote Leivant [22],

> we pursue a dual approach: rather than enriching the type systems to match logic, we impoverish logic to match the type structure.

What is remarkable is that even after this impoverishment enough power remains to capture the naturals and other algebraic types.

1.2. Introduction to the second edition

This paper has previously appeared twice, in a conference and a journal [47,48]. What justifies a third outing? The earlier versions used a logic over untyped lambda terms, similar to that considered by Mairson [23] and Krivine and Parigot [20]. This "second edition" uses a logic over polymorphically typed lambda terms, similar to that considered by Plotkin and Abadi [33] and by Takeuti [44]. In the previous paper, I claimed it "appears straightforward" to transpose the results from an untyped to a typed logic. Having now performed the exercise, I can say it was straightforward but not trivial — getting the formulation right required some care. This new version differs in many details of presentation, and clarifies the structure of the proofs. In particular, this paper gives a sharper characterization of the connection between the inductive naturals and binary parametricity.

Previously, I observed:

> Girard's Representation Theorem requires a logic with untyped terms, since the whole point of the theorem is to demonstrate that functions defined in a language without types may be represented in a language with types.

I no longer believe that is correct. In the definition of the naturals given above, N, s, and z are uninterpreted symbols, requiring only that s have type $\mathsf{N} \to \mathsf{N}$ and z have type N – in short, the types impose no constraints other than those one finds in an algebraic specification with sorts. From this algebraic definition, one can then *derive* the representation, in which N, s, and z are as usual for Church numerals.

So there is no serious disadvantage in moving to a typed version, and some advantages. In the untyped setting, extensionality is problematic and the only model known to satisfy parametricity is a term model; while in the typed setting, extensionality is unproblematic and there are many models satisfying parametricity. The untyped model is

Syntax

Type variables	X, Y, Z
Types	$A, B, C \ ::= \ X \mid A \to B \mid \forall X.\, B$
Individual variables	x, y, z
Terms	$s, t, u \ ::= \ x^A \mid \lambda x^A.\, u \mid s\ t \mid \Lambda X.\, u \mid s\ A$

Rules

$$\dfrac{\begin{matrix}[x^A]\\ \vdots \\ u^B\end{matrix}}{(\lambda x^A.\, u)^{A \to B}}\ \to\text{-I}^x \qquad \dfrac{s^{A \to B} \qquad t^A}{(s\ t)^B}\ \to\text{-E}$$

$$\dfrac{u^B}{(\Lambda X.\, u)^{\forall X.\, B}}\ \forall\text{-I} \quad X \text{ does not escape} \qquad \dfrac{s^{\forall X.\, B}}{(s\ A)^{B[A/X]}}\ \forall\text{-E}$$

Reductions

$$\dfrac{\dfrac{\begin{matrix}[x^A]\\ \vdots \\ u^B\end{matrix}}{(\lambda x^A.\, u)^{A \to B}}\ \to\text{-I}^x \qquad \begin{matrix}\vdots \\ t^A\end{matrix}}{((\lambda x^A.\, u)\ t)^B}\ \to\text{-E} \quad \Rightarrow \quad \begin{matrix}\vdots \\ u[t/x]^B\end{matrix} \qquad \dfrac{\dfrac{\begin{matrix}\vdots \\ u^B\end{matrix}}{(\Lambda X.\, u)^{\forall X.\, B}}\ \forall\text{-I}}{((\Lambda X.\, u)\ A)^{B[A/X]}}\ \forall\text{-E} \quad \Rightarrow \quad \begin{matrix}\vdots \\ u[A/X]^{B[A/X]}\end{matrix}$$

Fig. 1. Second-order lambda calculus, F2.

inconsistent if one assumes parametricity at all types (as shown in the first edition [47,48]), while the typed model is consistent if one assumes parametricity at all types (as shown by Takeuti [44]).

This version is an extensive rewrite of the previous versions. Two changes are worth particular note. First, previous versions use a sequent presentation of natural deduction, where all the hypotheses of a formula are written to the left of a turnstyle; while this version reverts to the original format of Gentzen [13], where only new hypotheses are written, eliminating much repetition. (I am grateful to Clemens Szyperski for inspiring this step, by asking "How can one get to the meat in inference rules?") Second, this version is in colour.

The remainder of this paper is organized as follows. Section 2 introduces the second-order lambda calculus F2 and the second-order logic P2. Section 3 describes the Girard projection and the Reynolds embedding. Section 4 explains doubling and parametricity. Section 5 explores the relation between induction and parametricity. Section 6 applies a realizability interpretation to prove Girard's Representation Theorem.

2. Systems F2 and P2

The second-order lambda calculus F2 is summarized in Fig. 1, and second-order intuitionistic logic P2 is summarized in Fig. 2.

Derivations of typings in F2 and proofs of propositions in P2 are written in the natural deduction style of Gentzen [13], with hypotheses in square brackets. Write $\equiv$ for syntactic equivalence of terms, propositions, predicates, derivations, or proofs, and write $\Rightarrow$ for reductions between terms, propositions, derivations, or proofs.

Related concepts are denoted with the same letters, using fonts to distinguish syntactic categories. Let x, y, z range over individual variables in F2 and P2 and bold $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ range over labels of hypotheses in P2. Let t, u, v range over terms in F2 and P2 and bold $\boldsymbol{s}, \boldsymbol{t}, \boldsymbol{u}$ range over proofs in P2. Let X, Y, Z range over type variables in F2 and P2, and calligraphic $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ range over predicate variables in P2. Let A, B, C range over types in F2 and P2, bold $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ range over propositions in P2, and calligraphic $\mathcal{A}, \mathcal{B}, \mathcal{C}$ range over predicates in P2.

Syntax

Hypothesis labels	x, y, z
Proofs	s, t, u
Predicate variables	$\mathcal{X}, \mathcal{Y}$
Propositions	$A, B \ ::= \ t^C \in \mathcal{A}^C \mid A \to B \mid \forall \mathcal{X}^C.\, B \mid \forall x^C.\, B \mid \forall X.\, B$
Predicates	$\mathcal{A}, \mathcal{B} \ ::= \ \mathcal{X}^C \mid \{x^C \mid A\}$

Rules

$$\dfrac{\begin{matrix}[A]^x \\ \vdots \\ B\end{matrix}}{A \to B}\ {\to}\text{-I}^x \qquad \dfrac{A \to B \qquad A}{B}\ {\to}\text{-E}$$

$$\dfrac{B}{\forall \mathcal{X}^C.\, B}\ \forall\text{-I} \quad \mathcal{X} \text{ does not escape} \qquad \dfrac{\forall \mathcal{X}^C.\, B}{B[\mathcal{A}^C/\mathcal{X}]}\ \forall\text{-E}$$

$$\dfrac{B}{\forall x^C.\, B}\ \forall^1\text{-I} \quad x \text{ does not escape} \qquad \dfrac{\forall x^C.\, B}{B[t^C/x]}\ \forall^1\text{-E}$$

$$\dfrac{B}{\forall X.\, B}\ \forall^2\text{-I} \quad X \text{ does not escape} \qquad \dfrac{\forall X.\, B}{B[A/X]}\ \forall^2\text{-E}$$

$$\dfrac{A}{B}\ \beta \quad A =_\beta B$$

Reductions

$$\dfrac{\dfrac{\begin{matrix}[A]^x \\ \vdots\, u \\ B\end{matrix}}{A \to B}\ {\to}\text{-I}^x \qquad \begin{matrix}\vdots\, t \\ A\end{matrix}}{B}\ {\to}\text{-E} \quad \Rightarrow \quad \begin{matrix}\vdots\, u[t/x] \\ B\end{matrix} \qquad \dfrac{\dfrac{\begin{matrix}\vdots\, u \\ B\end{matrix}}{\forall \mathcal{X}^C.\, B}\ \forall\text{-I}}{B[\mathcal{A}^C/\mathcal{X}]}\ \forall\text{-E} \quad \Rightarrow \quad \begin{matrix}\vdots\, u[\mathcal{A}^C/\mathcal{X}] \\ B[\mathcal{A}^C/\mathcal{X}]\end{matrix}$$

$$\dfrac{\dfrac{\begin{matrix}\vdots\, u \\ B\end{matrix}}{\forall x^C.\, B}\ \forall^1\text{-I}}{B[t^C/x]}\ \forall^1\text{-E} \quad \Rightarrow \quad \begin{matrix}\vdots\, u[t^C/x] \\ B[t^C/x]\end{matrix} \qquad \dfrac{\dfrac{\begin{matrix}\vdots\, u \\ B\end{matrix}}{\forall X.\, B}\ \forall^2\text{-I}}{B[A/X]}\ \forall^2\text{-E} \quad \Rightarrow \quad \begin{matrix}\vdots\, u[A/X] \\ B[A/X]\end{matrix}$$

Fig. 2. Second-order propositional logic, P2.

We write $u[t/x]$ for term u with term t substituted for individual variable x; and $B[A/X]$ for type B with type A substituted for type variable X; and $\boldsymbol{B}[\mathcal{A}/\mathcal{X}]$ for proposition $\boldsymbol{B}$ with predicate $\mathcal{A}$ substituted for predicate variable $\mathcal{X}$. All substitutions rename bound variables as necessary to avoid capture.

We superscript individual variables, terms, predicate variables, and predicates with their types. The grammar requires these superscripts to always be present, but in what follows they will be dropped when no confusion results.

Types are formed from type variables X, functions $A \to B$, and quantification over types $\forall X.\, B$. Terms are formed from individual variables x^A, abstraction $\lambda x^A.\, u$, application $s\ t$, type abstraction $\Lambda X.\, u$, and type application $s\ A$.

Each well-typed term t uniquely determines its typing derivation. We write

$$\begin{array}{c}\vdots \\ t^{A}\end{array}$$

to indicate that there is a derivation of term t of type A. Derivations have hypotheses of the form

$$[x^{A}]$$

to indicate that individual variable x has type A.

There are introduction and elimination rules for functions and quantification over types. In the introduction rule for functions, each instance of the rule is superscripted with the name x of the variable hypothesis that is discharged by that rule. In the introduction rule for quantifiers, the phrase "X does not escape" means that the type variable does not appear free in any undischarged hypothesis of the derivation.

A derivation reduces when an introduction rule is followed by an elimination rule for the same connective, corresponding to the usual β rules for term and type applications. We write $t =_{\beta} u$ if t and u can be shown equivalent by β reduction.

Propositions are formed from tests that a term satisfies a predicate $t^{C} \in \boldsymbol{\mathcal{A}}^{C}$, implication $\boldsymbol{A} \to \boldsymbol{B}$, quantification over individuals $\forall x^{C}.\, \boldsymbol{B}$, quantification over types $\forall X.\, \boldsymbol{B}$, and quantification over predicates $\boldsymbol{\forall \mathcal{X}^{C}.\, B}$. Predicates are formed from predicate variables $\boldsymbol{\mathcal{X}}^{C}$ and comprehensions $\{x^{C} \mid \boldsymbol{A}\}$; both of these are over individuals of type C. Here C is any type of F2, possibly including free type variables.

Unlike with typing derivations, proofs are not uniquely determined by their conclusions. We let $\boldsymbol{s}, \boldsymbol{t}, \boldsymbol{u}$ range over proofs, and write

$$\begin{array}{c}\vdots\, \boldsymbol{t} \\ \boldsymbol{A}\end{array}$$

to indicate that proof $\boldsymbol{t}$ concludes with proposition $\boldsymbol{A}$. Proofs have hypotheses of the form

$$[\boldsymbol{A}]^{\boldsymbol{x}}$$

where $\boldsymbol{A}$ is a proposition and $\boldsymbol{x}$ is a hypothesis label.

There are introduction and elimination rules for implication and quantification over individuals, types, and predicates. In the introduction rule for implication, each instance of the rule is superscripted with the label $\boldsymbol{x}$ of the hypothesis that is discharged by that rule. In the introduction rules for quantifiers, the phrase "$x/X/\boldsymbol{\mathcal{X}}$ does not escape" means that the individual/type/predicate variable does not appear free in any undischarged hypothesis of the derivation.

Beta equality on terms corresponds to the equivalent reductions for F2.

$$\begin{array}{l}(\lambda x^{T}.\, u)\; t \;=_{\beta}\; u[t/x] \\ (\Lambda X.\, u)\; A \;=_{\beta}\; u[A/X]\end{array}$$

Take $=_{\beta}$ to be the smallest congruence on propositions that respects the two equations above and one additional equation below.

Notationally, predicates are treated as sets, so $t^{C} \in \boldsymbol{\mathcal{A}}^{C}$ indicates that term t of type C satisfies predicate $\boldsymbol{\mathcal{A}}$ over individuals of type C, and $\{x^{C} \mid \boldsymbol{A}\}$ stands for the predicate over individual variable x of type C that holds when proposition $\boldsymbol{A}$ is satisfied. This gives us an additional notion of β equality between propositions.

$$t^{C} \in \{x^{C} \mid \boldsymbol{A}\} =_{\beta} \boldsymbol{A}[t/x]$$

(Some authors write $\boldsymbol{\mathcal{A}}\, t$ instead of $t \in \boldsymbol{\mathcal{A}}$, and $\lambda x.\, \boldsymbol{A}$ instead of $\{x \mid \boldsymbol{A}\}$.)

A proof reduces when an introduction rule is followed by an elimination rule for the same connective. We write $\boldsymbol{u}[\boldsymbol{t}/\boldsymbol{x}]$ to stand for the proof $\boldsymbol{u}$ with each occurrence of a hypothesis with label $\boldsymbol{x}$ is replaced by proof $\boldsymbol{t}$. We also write $\boldsymbol{u}[t/x]$, $\boldsymbol{u}[A/X]$, and $\boldsymbol{u}[\boldsymbol{\mathcal{A}}/\boldsymbol{\mathcal{X}}]$ to stand for proofs with substitution of a term for an individual variable, a types for a type variables, or a predicate for a predicate variable.

In addition to the listed reductions, we also have commuting conversions for (β). Here is a commuting conversion that pushes a use of (β) from the principle formula in $(\rightarrow$-E) down to the conclusion. Assume $\boldsymbol{B} =_\beta \boldsymbol{B}'$.

$$\dfrac{\dfrac{\dfrac{\vdots}{\boldsymbol{A} \rightarrow \boldsymbol{B}}}{\boldsymbol{A} \rightarrow \boldsymbol{B}'}\,\beta \qquad \dfrac{\vdots}{\boldsymbol{A}}}{\boldsymbol{B}'}\,\rightarrow\text{-}\mathbf{E} \quad \Rightarrow \quad \dfrac{\dfrac{\dfrac{\vdots}{\boldsymbol{A} \rightarrow \boldsymbol{B}} \qquad \dfrac{\vdots}{\boldsymbol{A}}}{\boldsymbol{B}}\,\rightarrow\text{-}\mathbf{E}}{\boldsymbol{B}'}\,\beta$$

There are similar conversions for each introduction and elimination rule. These commuting conversions are required if the Reynolds embedding is to preserves reductions, and they will be referred to in Proposition 1.

We can simulate proposition variables (nullary predicates) and relation variables (binary predicates) by defining unit types and pair types in the term calculus.

$$\begin{array}{lcl}
\mathsf{1} & \equiv & \forall X.\, X \rightarrow X \\
* & \equiv & \Lambda X.\, \lambda x^X.\, x \\
\{* \mid \boldsymbol{C}\} & \equiv & \{z^{\mathsf{1}} \mid \boldsymbol{C}\} \\
\\
A \times B & \equiv & \forall X.\, (A \rightarrow B \rightarrow X) \rightarrow X \\
(t^A, u^B) & \equiv & \Lambda X.\, \lambda k^{A \rightarrow B \rightarrow X}.\, k\ t\ u \\
\mathsf{fst}(s^{A\times B}) & \equiv & s\ A\ (\lambda x^A.\, \lambda y^B.\, x) \\
\mathsf{snd}(s^{A\times B}) & \equiv & s\ B\ (\lambda x^A.\, \lambda y^B.\, y) \\
\{(x^A, y^B) \mid \boldsymbol{C}\} & \equiv & \{z^{A\times B} \mid \boldsymbol{C}[\mathsf{fst}(z)/x, \mathsf{snd}(z)/y]\}
\end{array}$$

(Here z should not appear free in $\boldsymbol{A}$.) As required, we have

$$\begin{array}{lcl}
* \in \{* \mid \boldsymbol{A}\} & =_\beta & \boldsymbol{A} \\
(t, u) \in \{(x^A, y^B) \mid \boldsymbol{A}\} & =_\beta & \boldsymbol{A}[t/x, u/y]
\end{array}$$

True, false, conjunction, disjunction, and equivalence can be defined in terms of the connectives already given.

$$\begin{array}{lcl}
\top & \equiv & \forall \mathcal{X}^{\mathsf{1}}.\, * \in \mathcal{X} \rightarrow * \in \mathcal{X} \\
\bot & \equiv & \forall \mathcal{X}^{\mathsf{1}}.\, * \in \mathcal{X} \\
\boldsymbol{A} \wedge \boldsymbol{B} & \equiv & \forall \mathcal{X}^{\mathsf{1}}.\, (\boldsymbol{A} \rightarrow \boldsymbol{B} \rightarrow * \in \mathcal{X}) \rightarrow * \in \mathcal{X} \\
\boldsymbol{A} \vee \boldsymbol{B} & \equiv & \forall \mathcal{X}^{\mathsf{1}}.\, (\boldsymbol{A} \rightarrow * \in \mathcal{X}) \rightarrow (\boldsymbol{B} \rightarrow * \in \mathcal{X}) \rightarrow * \in \mathcal{X} \\
\boldsymbol{A} \leftrightarrow \boldsymbol{B} & \equiv & (\boldsymbol{A} \rightarrow \boldsymbol{B}) \wedge (\boldsymbol{B} \rightarrow \boldsymbol{A})
\end{array}$$

(Here $\mathcal{X}$ should not appear free in $\boldsymbol{A}$ or $\boldsymbol{B}$.)

Implication and equivalence of predicates are written with the usual set-theoretic notation.

$$\begin{array}{lcl}
\mathcal{A} \subseteq \mathcal{B} & \equiv & \forall x.\, x \in \mathcal{A} \rightarrow x \in \mathcal{B} \\
\mathcal{A} = \mathcal{B} & \equiv & (\mathcal{A} \subseteq \mathcal{B}) \wedge (\mathcal{B} \subseteq \mathcal{A})
\end{array}$$

Remarkably, P2 is powerful enough to express equality between terms. Following Leibniz, two terms are equal if every predicate that holds of the first also holds of the second.

$$t^A = u^A \equiv \forall \mathcal{X}^A.\, t \in \mathcal{X} \rightarrow u \in \mathcal{X}$$

It is easy to see that equality is reflexive.

$$\begin{array}{ll}
 & t = t \\
\equiv & \quad \text{defn} \\
 & \forall \mathcal{X}.\, t \in \mathcal{X} \rightarrow t \in \mathcal{X}
\end{array}$$

It is more subtle to see that it is symmetric.

$$\begin{array}{ll} & t = u \\ \equiv & \quad \text{defn} \\ & \forall \mathcal{X}.\, t \in \mathcal{X} \to u \in \mathcal{X} \\ \to & \quad \text{instantiate } \{x \mid x = t\}/\mathcal{X} \\ & t = t \to u = t \\ \equiv & \quad \text{equality is reflexive} \\ & u = t \end{array}$$

One may similarly show transitivity, and that $t =_\beta u$ implies $t = u$.

Extensionality is given by two families of axioms.

$$\forall f^{A \to B}, g^{A \to B}.\, (\forall x^A.\, f\ x = g\ x) \to f = g$$
$$\forall f^{\forall X.\, B}, g^{\forall X.\, B}.\, (\forall X.\, f\ X = g\ X) \to f = g$$

3. The Girard projection and the Reynolds embedding

The Girard projection takes a proposition $\boldsymbol{A}$ into a type $\boldsymbol{A}^\circ$ and a proof $\boldsymbol{t}$ into a term $\boldsymbol{t}^\circ$ such that

$$\left(\begin{array}{c} \vdots\, \boldsymbol{t} \\ \boldsymbol{A} \end{array}\right)^\circ \equiv \begin{array}{c} \vdots \\ \boldsymbol{t}^\circ \boldsymbol{A}^\circ \end{array}.$$

The Girard projection is defined in Fig. 3. It maps implication in P2 into function types in F2, quantification over predicates in P2 into quantification over types in F2, and discards quantification over individuals and types in P2. The Girard projection also takes a predicate $\boldsymbol{\mathcal{A}}^C$ into a type $\boldsymbol{\mathcal{A}}^\circ$. It maps a predicate variable into the corresponding type variable, and a comprehension into the Girard projection of the corresponding proposition.

The Reynolds embedding takes a type A into a predicate A^{*A}, and a term t into a proof t^* such that

$$\left(\begin{array}{c} \vdots \\ t^A \end{array}\right)^* \equiv \begin{array}{c} \vdots\, t^* \\ t \in A^* \end{array}.$$

The Reynolds embedding is defined in Fig. 4. It expands functions in F2 into quantification over individuals and into implication in P2, and quantification over types in F2 into quantification over types and predicates in P2.

It is easy to check that the Girard projection and Reynolds embedding preserve substitution, $(\boldsymbol{B}[\boldsymbol{\mathcal{A}}/\boldsymbol{\mathcal{X}}])^\circ \equiv \boldsymbol{B}^\circ[\boldsymbol{\mathcal{A}}^\circ/X]$ and $(B[A/X])^* \equiv B^*[A^*/\boldsymbol{\mathcal{X}}]$, and that the Girard projection is invariant under β reduction, if $\boldsymbol{A} =_\beta \boldsymbol{B}$ then $\boldsymbol{A}^\circ \equiv \boldsymbol{B}^\circ$.

Proposition 1 (*The Girard Projection and Reynolds Embedding Preserve Reduction*).

$$\left(\begin{array}{c} \vdots\, \boldsymbol{t} \\ \boldsymbol{A} \end{array} \Rightarrow \begin{array}{c} \vdots\, \boldsymbol{u} \\ \boldsymbol{A} \end{array}\right)^\circ \equiv \begin{array}{c} \vdots \\ \boldsymbol{t}^\circ \boldsymbol{A}^\circ \end{array} \Rightarrow \begin{array}{c} \vdots \\ \boldsymbol{u}^\circ \boldsymbol{A}^\circ \end{array} \qquad \left(\begin{array}{c} \vdots \\ t^A \end{array} \Rightarrow \begin{array}{c} \vdots \\ u^B \end{array}\right)^* \equiv \begin{array}{c} \vdots\, t^* \\ t \in A^* \end{array} \Rightarrow \dfrac{\begin{array}{c} \vdots\, u^* \\ u \in A^* \end{array}}{t \in A^*}\,\beta.$$

(An earlier version of this paper [47] failed to note the role of β rules in the preservation of reductions for the Reynolds embedding.)

Proof. The Girard projection takes reduction of an implication into reduction of a function; takes reduction of quantification over predicates into reduction of quantification over types; and removes reduction of quantification over individuals or types. The Reynolds embedding takes reduction of a function into reduction of a quantification over individuals followed by reduction of an implication, with a β rule remaining; and takes reduction of quantification over types into reduction of quantification over types followed by reduction of quantification over predicates, with a β rule remaining. The commuting rules for β described in the previous section may be used to push remaining β rules down to the conclusion of the proof. □

It is easy to see that the Reynolds embedding followed by the Girard projection is the identity.

Propositions

$$
\begin{array}{lcl}
(t^C \in \boldsymbol{\mathcal{A}}^C)^\circ & \equiv & \boldsymbol{\mathcal{A}}^\circ \\
(\boldsymbol{A} \to \boldsymbol{B})^\circ & \equiv & \boldsymbol{A}^\circ \to \boldsymbol{B}^\circ \\
(\forall \boldsymbol{\mathcal{X}}^C .\, \boldsymbol{B})^\circ & \equiv & \forall X.\, \boldsymbol{B}^\circ \\
(\forall x^C .\, \boldsymbol{B})^\circ & \equiv & \boldsymbol{B}^\circ \\
(\forall X.\, \boldsymbol{B})^\circ & \equiv & \boldsymbol{B}^\circ
\end{array}
$$

Predicates

$$
\begin{array}{lcl}
(\boldsymbol{\mathcal{X}}^C)^\circ & \equiv & X \\
(\{x^C \mid \boldsymbol{A}\})^\circ & \equiv & \boldsymbol{A}^\circ
\end{array}
$$

Proofs

$$
\left(\dfrac{\begin{array}{c}[\boldsymbol{A}]^x \\ \vdots\, \boldsymbol{u} \\ \boldsymbol{B}\end{array}}{\boldsymbol{A} \to \boldsymbol{B}}\ \boldsymbol{\to}\textbf{-I}^x\right)^\circ \equiv \dfrac{\begin{array}{c}[x^{\boldsymbol{A}^\circ}] \\ \vdots \\ \boldsymbol{u}^{\circ \boldsymbol{B}^\circ}\end{array}}{(\lambda x^{\boldsymbol{A}^\circ}.\, \boldsymbol{u}^\circ)^{\boldsymbol{A}^\circ \to \boldsymbol{B}^\circ}}\ \to\text{-I}^x
\qquad
\left(\dfrac{\begin{array}{c}\vdots\, \boldsymbol{s} \\ \boldsymbol{A} \to \boldsymbol{B}\end{array} \quad \begin{array}{c}\vdots\, \boldsymbol{t} \\ \boldsymbol{A}\end{array}}{\boldsymbol{B}}\ \boldsymbol{\to}\textbf{-E}\right)^\circ \equiv \dfrac{\begin{array}{c}\vdots \\ \boldsymbol{s}^{\circ \boldsymbol{A}^\circ \to \boldsymbol{B}^\circ}\end{array} \quad \begin{array}{c}\vdots \\ \boldsymbol{t}^{\circ \boldsymbol{A}^\circ}\end{array}}{(\boldsymbol{s}^\circ\ \boldsymbol{t}^\circ)^{\boldsymbol{B}^\circ}}\ \to\text{-E}
$$

$$
\left(\dfrac{\begin{array}{c}\vdots\, \boldsymbol{u} \\ \boldsymbol{B}\end{array}}{\forall \boldsymbol{\mathcal{X}}^C .\, \boldsymbol{B}}\ \boldsymbol{\forall}\textbf{-I}\right)^\circ \equiv \dfrac{\begin{array}{c}\vdots \\ \boldsymbol{u}^{\circ \boldsymbol{B}^\circ}\end{array}}{(\Lambda X.\, \boldsymbol{u}^\circ)^{\forall X.\, \boldsymbol{B}^\circ}}\ \forall\text{-I}
\qquad
\left(\dfrac{\begin{array}{c}\vdots\, \boldsymbol{s} \\ \forall \boldsymbol{\mathcal{X}}^C .\, \boldsymbol{B}\end{array}}{\boldsymbol{B}[\boldsymbol{\mathcal{A}}^C / \boldsymbol{\mathcal{X}}]}\ \boldsymbol{\forall}\textbf{-E}\right)^\circ \equiv \dfrac{\begin{array}{c}\vdots \\ \boldsymbol{s}^{\circ \forall X.\, \boldsymbol{B}^\circ}\end{array}}{(\boldsymbol{s}^\circ\ \boldsymbol{\mathcal{A}}^\circ)^{\boldsymbol{B}^\circ[\boldsymbol{\mathcal{A}}^\circ / X]}}\ \forall\text{-E}
$$

$$
\left(\dfrac{\begin{array}{c}\vdots\, \boldsymbol{u} \\ \boldsymbol{B}\end{array}}{\forall x^C .\, \boldsymbol{B}}\ \forall^1\text{-I}\right)^\circ \equiv \begin{array}{c}\vdots \\ \boldsymbol{u}^{\circ \boldsymbol{B}^\circ}\end{array}
\qquad
\left(\dfrac{\begin{array}{c}\vdots\, \boldsymbol{s} \\ \forall x^C .\, \boldsymbol{B}\end{array}}{\boldsymbol{B}[t^C / x]}\ \forall^1\text{-E}\right)^\circ \equiv \begin{array}{c}\vdots \\ \boldsymbol{s}^{\circ \boldsymbol{B}^\circ}\end{array}
$$

$$
\left(\dfrac{\begin{array}{c}\vdots\, \boldsymbol{u} \\ \boldsymbol{B}\end{array}}{\forall X.\, \boldsymbol{B}}\ \forall^2\text{-I}\right)^\circ \equiv \begin{array}{c}\vdots \\ \boldsymbol{u}^{\circ \boldsymbol{B}^\circ}\end{array}
\qquad
\left(\dfrac{\begin{array}{c}\vdots\, \boldsymbol{s} \\ \forall X.\, \boldsymbol{B}\end{array}}{\boldsymbol{B}[A / X]}\ \forall^2\text{-E}\right)^\circ \equiv \begin{array}{c}\vdots \\ \boldsymbol{s}^{\circ \boldsymbol{B}^\circ}\end{array}
$$

$$
\left(\dfrac{\begin{array}{c}\vdots\, \boldsymbol{t} \\ \boldsymbol{A}\end{array}}{\boldsymbol{B}}\ \beta\right)^\circ \equiv \begin{array}{c}\vdots \\ \boldsymbol{t}^{\circ \boldsymbol{A}^\circ}\end{array}
$$

Fig. 3. The Girard projection.

Proposition 2 (*Girard Inverts Reynolds*).

$$
A^{*\circ} \equiv A \qquad\qquad \left(\begin{array}{c}\vdots \\ t^A\end{array}\right)^{*\circ} \equiv \begin{array}{c}\vdots \\ t^A\end{array}.
$$

For example, here is the type of the Church numerals in F2.

$$
\mathsf{N} \equiv \forall X.\, (X \to X) \to X \to X
$$

Applying the Reynolds embedding yields the following predicate in P2.

$$
\mathsf{N}^* \equiv \{n^{\mathsf{N}} \mid \forall X.\, \boldsymbol{\forall} \boldsymbol{\mathcal{X}}^X .\, \forall s^{X \to X}.\, (\forall m^X.\, m \in \boldsymbol{\mathcal{X}} \boldsymbol{\to} s\ m \in \boldsymbol{\mathcal{X}}) \boldsymbol{\to} \forall z^X.\, z \in \boldsymbol{\mathcal{X}} \boldsymbol{\to} n\ X\ s\ z \in \boldsymbol{\mathcal{X}}\}
$$

Types

$$\begin{array}{lcl} (X)^* & \equiv & \boldsymbol{\mathcal{X}}^X \\ (A \to B)^* & \equiv & \{z^{A\to B} \mid \forall x^A.\, x \in A^* \to z\,x \in B^*\} \\ (\forall X.\,B)^* & \equiv & \{z^{\forall X.\,B} \mid \forall X.\,\boldsymbol{\forall}\boldsymbol{\mathcal{X}}^X.\, z\,X \in B^*\}\end{array}$$

Proofs

$$\left(\dfrac{\begin{array}{c}[x^A]\\ \vdots \\ u^B\end{array}}{(\lambda x^A.\,u)^{A\to B}}\to\text{-I}^x\right)^* \equiv \dfrac{\dfrac{\dfrac{\begin{array}{c}[x\in A^*]^x\\ \vdots\, u^* \\ u \in B^*\end{array}}{(\lambda x^A.\,u)\,x \in B^*}\,\beta}{x\in A^* \to (\lambda x^A.\,u)\,x\in B^*}\,\boldsymbol{\to}\textbf{-I}^x}{\forall x^A.\,x\in A^*\to(\lambda x^A.\,u)\,x\in B^*}\,\forall^1\text{-I}$$

$$\left(\dfrac{\begin{array}{c}\vdots\\ s^{A\to B}\end{array} \quad \begin{array}{c}\vdots\\ t^A\end{array}}{(s\,t)^B}\to\text{-E}\right)^* \equiv \dfrac{\dfrac{\begin{array}{c}\vdots\, s^*\\ \forall x^A.\,x\in A^*\to s\,x\in B^*\end{array}}{t\in A^*\to s\,t\in B^*}\,\forall^1\text{-E} \quad \begin{array}{c}\vdots\, t^*\\ t\in A^*\end{array}}{s\,t\in B^*}\,\boldsymbol{\to}\textbf{-E}$$

$$\left(\dfrac{\begin{array}{c}\vdots\\ u^B\end{array}}{(\Lambda X.\,u)^{\forall X.\,B}}\forall\text{-I}\right)^* \equiv \dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\, u^*\\ u\in B^*\end{array}}{(\Lambda X.\,u)\,X\in B^*}\,\beta}{\boldsymbol{\forall}\boldsymbol{\mathcal{X}}^X.\,(\Lambda X.\,u)\,X\in B^*}\,\boldsymbol{\forall}\textbf{-I}}{\forall X.\,\boldsymbol{\forall}\boldsymbol{\mathcal{X}}^X.\,(\Lambda X.\,u)\,X\in B^*}\,\forall^2\text{-I}$$

$$\left(\dfrac{\begin{array}{c}\vdots\\ s^{\forall X.\,B}\end{array}}{(s\;A)^{B[A/X]}}\forall\text{-E}\right)^* \equiv \dfrac{\dfrac{\begin{array}{c}\vdots\, s^*\\ \forall X.\,\boldsymbol{\forall}\boldsymbol{\mathcal{X}}^X.\,s\,X\in B^*\end{array}}{\boldsymbol{\forall}\boldsymbol{\mathcal{X}}^A.\,s\,A\in B^*[A/X]}\,\forall^2\text{-E}}{s\,A\in B^*[A/X,\,A^*/\boldsymbol{\mathcal{X}}]}\,\boldsymbol{\forall}\textbf{-E}$$

Fig. 4. The Reynolds embedding.

Applying the Girard projection then yields the original type.

$$\mathsf{N}^{*\circ} \equiv \forall X.\,(X \to X) \to X \to X$$

Define $\mathsf{2}^{\mathsf{N}} = \Lambda X.\,\lambda s^{X\to X}.\,\lambda z^X.\,s\,(s\,z)$. Then $\mathsf{2} \in \mathsf{N}^*$ and $\mathsf{2}^{*\circ} \equiv \mathsf{2}$.

$$\left(\begin{array}{c}\vdots\\ \mathsf{2}^{\mathsf{N}}\end{array}\right)^{*\circ} \equiv \left(\begin{array}{c}\vdots\, \mathsf{2}^*\\ \mathsf{2}\in \mathsf{N}^*\end{array}\right)^\circ \equiv \begin{array}{c}\vdots\\ \mathsf{2}^{\mathsf{N}}\end{array}$$

Note that the Girard projection takes equality into the unit type.

$$(t = u)^\circ \equiv (\boldsymbol{\forall}\boldsymbol{\mathcal{X}}.\,t\in\boldsymbol{\mathcal{X}}\to u\in\boldsymbol{\mathcal{X}})^\circ \equiv \forall X.\,X\to X \equiv \mathsf{1}$$

Hence, the Girard projection erases any information content in the proof of an equality judgement. Similarly, one may extend the Girard projection so that it maps the extensionality axioms into the identity function at the unit type, $(\lambda x^{\mathsf{1}}.\,x)^{\mathsf{1}\to\mathsf{1}}$.

4. Doubling and parametricity

4.1. Doubling

The Reynolds embedding corresponds to a unary version of Reynolds's Abstraction Theorem. We can recover the binary version by means of a *doubling* mapping from P2 to P2.

Doubling is defined with the aid of operations that rename variables. For each individual variable x there is a renaming x', and for each type variable X there is a renaming X'. We write t' for the term that results from renaming all the free individual and type variables in t.

Doubling takes a proposition $\boldsymbol{A}$ into a proposition $\boldsymbol{A}^{\ddagger}$, a predicate $\boldsymbol{\mathcal{A}}^{C}$ into a predicate $\boldsymbol{\mathcal{A}}^{\ddagger\, C\times C'}$, and a proof $\boldsymbol{t}$ into a proof $\boldsymbol{t}^{\ddagger}$ such that

$$\left(\begin{array}{c}\vdots\ \boldsymbol{u}\\ \boldsymbol{B}\end{array}\right)^{\ddagger} \equiv \begin{array}{c}\vdots\ \boldsymbol{u}^{\ddagger}\\ \boldsymbol{B}^{\ddagger}\end{array}.$$

Doubling is defined in Fig. 5. It maps implication into itself, quantification over unary predicates into quantification over binary predicates, and quantifications over individuals and types into pairs of quantifications.

Doubling preserves substitution and β equality, $(\boldsymbol{B}[\boldsymbol{\mathcal{A}}^{C}/\boldsymbol{\mathcal{X}}^{C}])^{\ddagger} \equiv \boldsymbol{B}^{\ddagger}[\boldsymbol{\mathcal{A}}^{\ddagger\, C\times C'}/\boldsymbol{\mathcal{X}}^{C\times C'}]$, and if $\boldsymbol{A} =_{\beta} \boldsymbol{B}$ then $\boldsymbol{A}^{\ddagger} \equiv \boldsymbol{B}^{\ddagger}$.

Like the Girard projection and the Reynolds embedding, doubling is a homomorphism.

Proposition 3 (*Doubling Preserves Reductions*).

$$\left(\begin{array}{c}\vdots\ \boldsymbol{t}\\ \boldsymbol{A}\end{array} \Rightarrow \begin{array}{c}\vdots\ \boldsymbol{u}\\ \boldsymbol{A}\end{array}\right)^{\ddagger} \equiv \begin{array}{c}\vdots\ \boldsymbol{t}^{\ddagger}\\ \boldsymbol{A}^{\ddagger}\end{array} \Rightarrow \begin{array}{c}\vdots\ \boldsymbol{u}^{\ddagger}\\ \boldsymbol{A}^{\ddagger}\end{array}.$$

What Reynolds calls the Abstraction Theorem [36] and what Plotkin and Abadi call the Logical Relations Lemma [33] arises as the composition of Reynolds embedding with doubling.

Proposition 4 (*Abstraction Theorem*).

$$\left(\begin{array}{c}\vdots\\ t^{A}\end{array}\right)^{*\ddagger} \equiv \begin{array}{c}\vdots\ t^{*\ddagger}\\ (t, t') \in A^{*\ddagger}\end{array}.$$

In other places, the statement of the Abstraction Theorem must explicitly mention that the free variables of term t must satisfy logical relations corresponding to their types; this is implicit in the notation adopted here. The proof of the theorem is implicit in its statement, as it follows immediately from the definitions of the Reynolds embedding and doubling.

Here again is the type of the Church numerals in F2.

$$\mathsf{N} \equiv \forall X.\,(X \to X) \to X \to X$$

Applying the Reynolds embedding followed by doubling yields the following predicate in P2.

$$\begin{aligned}\mathsf{N}^{*\ddagger} \equiv \{(n^{\mathsf{N}}, n'^{\mathsf{N}}) \mid \forall X.\,\forall X'.\,\boldsymbol{\forall\mathcal{X}}^{X\times X'}.\ & \\ \forall s^{X\to X}.\,\forall s'^{X'\to X'}.\,&(\forall m^{X}.\,\forall m'^{X'}.\,(m, m') \in \boldsymbol{\mathcal{X}} \boldsymbol{\to} (s\ m, s'\ m') \in \boldsymbol{\mathcal{X}}) \boldsymbol{\to} \\ &\forall z^{X}.\,\forall z'^{X'}.\,(z, z') \in \boldsymbol{\mathcal{X}} \boldsymbol{\to} (n\ X\ s\ z, n'\ X'\ s'\ z') \in \boldsymbol{\mathcal{X}}\}\end{aligned}$$

Define $\mathsf{2}^{\mathsf{N}} = \Lambda X.\,\lambda s^{X\to X}.\,\lambda z^{X}.\,s\ (s\ z)$. Then $(\mathsf{2}, \mathsf{2}) \in \mathsf{N}^{*\ddagger}$.

$$\left(\begin{array}{c}\vdots\\ \mathsf{2}^{\mathsf{N}}\end{array}\right)^{*\ddagger} \equiv \left(\begin{array}{c}\vdots\ \mathsf{2}^{*}\\ \mathsf{2} \in \mathsf{N}^{*}\end{array}\right)^{\ddagger} \equiv \begin{array}{c}\vdots\ \mathsf{2}^{*\ddagger}\\ (\mathsf{2}, \mathsf{2}) \in \mathsf{N}^{*\ddagger}\end{array}$$

Propositions

$$
\begin{array}{lcl}
(t^C \in \mathcal{A}^C)^{\ddagger} & \equiv & (t^C, t'^{C'}) \in \mathcal{A}^{\ddagger C \times C'} \\
(A \to B)^{\ddagger} & \equiv & A^{\ddagger} \to B^{\ddagger} \\
(\forall \mathcal{X}^C.\, B)^{\ddagger} & \equiv & \forall \mathcal{X}^{C \times C'}.\, B^{\ddagger} \\
(\forall x^C.\, B)^{\ddagger} & \equiv & \forall x^C, x'^{C'}.\, B^{\ddagger} \\
(\forall X.\, B)^{\ddagger} & \equiv & \forall X, X'.\, B^{\ddagger}
\end{array}
$$

Predicates

$$
\begin{array}{lcl}
(\mathcal{X}^C)^{\ddagger} & \equiv & \mathcal{X}^{C \times C'} \\
(\{x^C \mid A\})^{\ddagger} & \equiv & \{(x^C, x'^{C'}) \mid A^{\ddagger}\}
\end{array}
$$

Proofs

$$
\left(\dfrac{\begin{array}{c}[A]^x \\ \vdots\, u \\ B\end{array}}{A \to B}\ \to\text{-I}^x \right)^{\ddagger} \equiv \dfrac{\begin{array}{c}[A^{\ddagger}]^x \\ \vdots\, u^{\ddagger} \\ B^{\ddagger}\end{array}}{A^{\ddagger} \to B^{\ddagger}}\ \to\text{-I}^x
\qquad
\left(\dfrac{\begin{array}{c}\vdots\, s \\ A \to B\end{array} \quad \begin{array}{c}\vdots\, t \\ A\end{array}}{B}\ \to\text{-E} \right)^{\ddagger} \equiv \dfrac{\begin{array}{c}\vdots\, s^{\ddagger} \\ A^{\ddagger} \to B^{\ddagger}\end{array} \quad \begin{array}{c}\vdots\, t^{\ddagger} \\ A^{\ddagger}\end{array}}{B^{\ddagger}}\ \to\text{-E}
$$

$$
\left(\dfrac{\begin{array}{c}\vdots\, u \\ B\end{array}}{\forall \mathcal{X}^C.\, B}\ \forall\text{-I} \right)^{\ddagger} \equiv \dfrac{\begin{array}{c}\vdots\, u^{\ddagger} \\ B^{\ddagger}\end{array}}{\forall \mathcal{X}^{C \times C'}.\, B^{\ddagger}}\ \forall\text{-I}
\qquad
\left(\dfrac{\begin{array}{c}\vdots\, s \\ \forall \mathcal{X}^C.\, B\end{array}}{B[\mathcal{A}^C/\mathcal{X}]}\ \forall\text{-E} \right)^{\ddagger} \equiv \dfrac{\begin{array}{c}\vdots\, s^{\ddagger} \\ \forall \mathcal{X}^{C \times C'}.\, B^{\ddagger}\end{array}}{B^{\ddagger}[\mathcal{A}^{\ddagger C \times C'}/\mathcal{X}]}\ \forall\text{-E}
$$

$$
\left(\dfrac{\begin{array}{c}\vdots\, u \\ B\end{array}}{\forall x^C.\, B}\ \forall^1\text{-I} \right)^{\ddagger} \equiv \dfrac{\begin{array}{c}\vdots\, u^{\ddagger} \\ B^{\ddagger}\end{array}}{\forall x^C, x'^{C'}.\, B^{\ddagger}}\ \forall^1\text{-I twice}
\qquad
\left(\dfrac{\begin{array}{c}\vdots\, s \\ \forall x^C.\, B\end{array}}{B[t^C/x]}\ \forall^1\text{-E} \right)^{\ddagger} \equiv \dfrac{\begin{array}{c}\vdots\, s^{\ddagger} \\ \forall x^C, x'^{C'}.\, B^{\ddagger}\end{array}}{B^{\ddagger}[t^C/x, t'^{C'}/x']}\ \forall^1\text{-E twice}
$$

$$
\left(\dfrac{\begin{array}{c}\vdots\, u \\ B\end{array}}{\forall X.\, B}\ \forall^2\text{-I} \right)^{\ddagger} \equiv \dfrac{\begin{array}{c}\vdots\, u^{\ddagger} \\ B^{\ddagger}\end{array}}{\forall X, X'.\, B^{\ddagger}}\ \forall^2\text{-I twice}
\qquad
\left(\dfrac{\begin{array}{c}\vdots\, s \\ \forall X.\, B\end{array}}{B[A/X]}\ \forall^2\text{-E} \right)^{\ddagger} \equiv \dfrac{\begin{array}{c}\vdots\, s^{\ddagger} \\ \forall X, X'.\, B^{\ddagger}\end{array}}{B^{\ddagger}[A/X, A'/X']}\ \forall^2\text{-E twice}
$$

$$
\left(\dfrac{\begin{array}{c}\vdots\, t \\ A\end{array}}{B}\ \beta \right)^{\ddagger} \equiv \dfrac{\begin{array}{c}\vdots\, t^{\ddagger} \\ A^{\ddagger}\end{array}}{B^{\ddagger}}\ \beta\text{ twice}
$$

Fig. 5. The doubling embedding.

4.2. *Parametricity*

It is convenient to introduce the notion of an identity relation at a type and at a predicate.

Definition 5. The *identity relation* at type A is defined by

$$A^{=} \equiv \{(x^A, x'^A) \mid x = x'\}.$$

Definition 6. The *identity relation* at a predicate $\mathcal{A}^A$ is defined by

$$\mathcal{A}^{=} \equiv \{(x^A, x'^A) \mid x = x' \wedge x \in \mathcal{A}\}.$$

The parametric closure of a type is the doubling of the Reynolds embedding of that type, with the relation corresponding to each free type variable taken to be the identity relation at that type.

Definition 7. The *parametric closure* on type A is defined by

$$A^{\approx} \equiv A^{*\ddagger}[X_1/X_1', X_1{}^{=}/\boldsymbol{\mathcal{X}}_1, \ldots, X_n/X_n', X_n{}^{=}/\boldsymbol{\mathcal{X}}_n]$$

where $X_1, \ldots, X_n$ are the free type variables in A.

It is easy to verify that $A^{\approx}$ is symmetric and transitive. If A is a closed type, then $A^{\approx} \equiv A^{*\ddagger}$, for example, $\mathsf{N}^{\approx} \equiv \mathsf{N}^{*\ddagger}$.

A type is *parametric* when all values of that type belong to the parametric closure, and is *extensive* when values are related by the parametric closure only if they are equal.

Definition 8. Type A is *parametric* when $A^{=} \subseteq A^{\approx}$ and *extensive* when $A^{\approx} \subseteq A^{=}$.

Equivalently, A is parametric when $\forall x^A.\, (x, x) \in A^{\approx}$ and extensive when $\forall x^A, x'^A.\, (x, x') \in A^{\approx} \to x = x'$. Note that we distinguish *extensiveness* as given here from *extensionality* as given at the end of Section 2.

Reynolds's *Parametricity Postulate* assumes that every quantified type is parametric. An immediate consequence is the following.

Proposition 9 (*Identity Extension Lemma*). *If every quantified type $\forall X.\, B$ is parametric, then every type is both parametric and extensive, $A^{\approx} = A^{=}$.*

Proof. The proof is by induction on the structure of types, and has five parts.

(i) Every type variable X is parametric and extensive. Immediate.

(ii) If A is parametric and B is extensive then $A \to B$ is extensive.

$$\begin{array}{ll} & (z, z') \in (A \to B)^{\approx} \\ \equiv & \quad \text{definition parametric closure} \\ & \forall x, x'.\, (x, x') \in (A)^{\approx} \to (z\, x, z'\, x') \in (B)^{\approx} \\ \to & \quad \text{induction hypothesis} \\ & \forall x, x'.\, x = x' \to z\, x = z'\, x' \\ \equiv & \quad \text{extensionality} \\ & z = z' \end{array}$$

(iii) If A is extensive and B is parametric then $A \to B$ is parametric. Similar to (ii).

(iv) If B is extensive then $\forall X.\, B$ is extensive. Let $X, \bar{Y}$ be the free variables of B.

$$\begin{array}{ll} & (z, z') \in (\forall X.\, B)^{\approx} \\ \equiv & \quad \text{definition parametric closure} \\ & \forall X, X'.\, \boldsymbol{\forall \mathcal{X}.}\, (z\, X, z'\, X') \in B^{*\ddagger}[\bar{Y}/\bar{Y}', \bar{Y}^{=}/\bar{\boldsymbol{\mathcal{Y}}}] \\ \to & \quad \text{instantiate } X/X', X^{=}/\boldsymbol{\mathcal{X}} \\ & \forall X.\, (z\, X, z'\, X) \in B^{*\ddagger}[X/X', X^{=}/\boldsymbol{\mathcal{X}}, \bar{Y}/\bar{Y}', \bar{Y}^{=}/\bar{\boldsymbol{\mathcal{Y}}}] \\ \to & \quad \text{induction hypothesis} \\ & \forall X.\, z\, X = z'\, X \\ \equiv & \quad \text{extensionality} \\ & z = z' \end{array}$$

(v) $\forall X.\, B$ is parametric. Given. □

Plotkin and Abadi [33] take parametricity at every quantified type and extensionality as axioms of their logic. Takeuti [44] takes parametricity and extensiveness at every type as axioms of his logic. It is not difficult to show these two sets of axioms are equivalent. Takeuti also shows that these assumptions are consistent, in that adding them to the logic does not permit derivation of the proposition false.

$$\left(\dfrac{\dfrac{\dfrac{\dfrac{[\mathsf{z} \in \mathcal{X}]^{z}}{\mathsf{z} \in \mathcal{X} \to \mathsf{z} \in \mathcal{X}}\;{\to}\text{-}\mathbf{I}^{z}}{(\forall m^{\mathsf{N}}.\, m \in \mathcal{X} \to \mathsf{s}\, m \in \mathcal{X}) \to \mathsf{z} \in \mathcal{X} \to \mathsf{z} \in \mathcal{X}}\;{\to}\text{-}\mathbf{I}^{s}}{\forall \mathcal{X}^{\mathsf{N}}.\, (\forall m^{\mathsf{N}}.\, m \in \mathcal{X} \to \mathsf{s}\, m \in \mathcal{X}) \to \mathsf{z} \in \mathcal{X} \to \mathsf{z} \in \mathcal{X}}\;\forall\text{-}\mathbf{I}}{\mathsf{z} \in \mathbf{N}}\;\beta\right)^{\circ}$$

$$\equiv \quad \dfrac{\dfrac{\dfrac{[z^{X}]}{\lambda z^{X}.\, z^{X \to X}}\;{\to}\text{-I}^{z}}{\lambda s^{X \to X}.\, \lambda z^{X}.\, z^{(X \to X) \to X \to X}}\;{\to}\text{-I}^{s}}{\Lambda X.\, \lambda s^{X \to X}.\, \lambda z^{X}.\, z^{\mathsf{N}}}\;\forall\text{-I}$$

Fig. 6. Zero.

However, we will make no use of the Parametricity Postulate or the Identity Extension Lemma in what follows. We assume extensionality, but not parametricity or extensiveness.

5. Parametricity is inductive

In this section we consider relations between parametricity and inductive definitions for the natural numbers. The results extend to any *data system* of the style considered by Böhm and Berarducci [4], Leivant [22], and Krivine and Parigot [20].

We consider two interpretations of the natural numbers, an *inductive* interpretation, $\mathbf{N}$, and a *deductive* interpretation N. The inductive interpretation $\mathbf{N}$ corresponds to the induction principle for natural numbers.

$$\mathbf{N} \equiv \{n \mid \forall X.\, \forall \mathcal{X}.\, (\forall m.\, m \in \mathcal{X} \to \mathsf{s}\, m \in \mathcal{X}) \to \mathsf{z} \in \mathcal{X} \to n \in \mathcal{X}\}$$

To prove a property of natural numbers by induction, one must show that for all m, if m has the property then its successor $\mathsf{s}\, m$ has the property, and one must show that z has the property. The above definition states that a value is a natural number if one can prove a property of it by induction. The idea of classifying induction principles using second-order propositional variables, and of defining a type via its induction principle, goes back to Frege [11].

One immediate consequence of the definition is that s and z do indeed construct natural numbers.

Proposition 10 (*Constructor Lemma*). *The following are provable in* P2.

$$n \in \mathbf{N} \to \mathsf{s}\, n \in \mathbf{N} \qquad\qquad \mathsf{z} \in \mathbf{N}$$

Proof. Straightforward. The proofs appear in the top parts of Figs. 6 and 7. □

The inductive interpretation and the Constructor Lemma do not depend in any way on the structure of N, s, and z, and they may be chosen to be uninterpreted constants. However, we will see there is good reason to choose N, s, and z to be their usual representations under the Church numerals, which we call the *deductive* interpretation.

$$\begin{array}{lll}
\mathsf{N} & \equiv & \forall X.\, (X \to X) \to (X \to X) \\
\mathsf{s}^{\mathsf{N} \to \mathsf{N}} & \equiv & \lambda m^{\mathsf{N}}.\, \Lambda X.\, \lambda s^{X \to X}.\, \lambda z^{X}.\, s\ (m\ s\ z) \\
\mathsf{z}^{\mathsf{N}} & \equiv & \Lambda X.\, \lambda s^{X \to X}.\, \lambda z^{X}.\, z
\end{array}$$

$$
\left(
\dfrac{
\dfrac{[\forall m^{\mathsf{N}}.\, m \in \mathcal{X} \to \mathsf{s}\, m \in \mathcal{X}]^{s}}{n \in \mathcal{X} \to \mathsf{s}\, n \in \mathcal{X}}\;\forall^{1}\text{-E}
\qquad
\dfrac{
\dfrac{
\dfrac{
\dfrac{[n \in \mathbf{N}]^{n}}{\forall \mathcal{X}^{\mathsf{N}}.\, (\forall m^{\mathsf{N}}.\, m \in \mathcal{X} \to \mathsf{s}\, m \in \mathcal{X}) \to \mathsf{z} \in \mathcal{X} \to n \in \mathcal{X}}\;\beta
}{(\forall m^{\mathsf{N}}.\, m \in \mathcal{X} \to \mathsf{s}\, m \in \mathcal{X}) \to \mathsf{z} \in \mathcal{X} \to n \in \mathcal{X}}\;\forall\text{-E}
\quad
[\forall m^{\mathsf{N}}.\, m \in \mathcal{X} \to \mathsf{s}\, m \in \mathcal{X}]^{s}
}{\mathsf{z} \in \mathcal{X} \to n \in \mathcal{X}}\;\to\text{-E}
\quad
[\mathsf{z} \in \mathcal{X}]^{z}
}{n \in \mathcal{X}}\;\to\text{-E}
}{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\mathsf{s}\, n \in \mathcal{X}
}{\mathsf{z} \in \mathcal{X} \to \mathsf{s}\, n \in \mathcal{X}}\;\to\text{-I}^{z}
}{(\forall m^{\mathsf{N}}.\, m \in \mathcal{X} \to \mathsf{s}\, m \in \mathcal{X}) \to \mathsf{z} \in \mathcal{X} \to \mathsf{s}\, n \in \mathcal{X}}\;\to\text{-I}^{s}
}{\forall \mathcal{X}^{\mathsf{N}}.\, (\forall m^{\mathsf{N}}.\, m \in \mathcal{X} \to \mathsf{s}\, m \in \mathcal{X}) \to \mathsf{z} \in \mathcal{X} \to \mathsf{s}\, n \in \mathcal{X}}\;\forall\text{-I}
}{\mathsf{s}\, n \in \mathbf{N}}\;\beta
}{
\dfrac{n \in \mathbf{N} \to \mathsf{s}\, n \in \mathbf{N}}{\forall n^{\mathsf{N}}.\, n \in \mathbf{N} \to \mathsf{s}\, n \in \mathbf{N}}\;\forall^{1}\text{-I}
}\;\to\text{-I}^{n}
}\;\to\text{-E}
\right)^{\circ}
$$

$$
\equiv\quad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
[s^{X\to X}]
\qquad
\dfrac{
\dfrac{
\dfrac{[n^{\mathsf{N}}]}{(n\ X)^{(X\to X)\to X\to X}}\;\forall\text{-E}
\quad
[s^{X\to X}]
}{(n\ X\ s)^{X\to X}}\;\to\text{-E}
\quad
[z^{X}]
}{(n\ X\ s\ z)^{X}}\;\to\text{-E}
}{(s\ (n\ X\ s\ z))^{X}}\;\to\text{-E}
}{(\lambda z^{X}.\, s\ (n\ X\ s\ z))^{X\to X}}\;\to\text{-I}^{z}
}{(\lambda s^{X\to X}.\, \lambda z^{X}.\, s\ (n\ X\ s\ z))^{(X\to X)\to X\to X}}\;\to\text{-I}^{s}
}{(\Lambda X.\, \lambda s^{X\to X}.\, \lambda z^{X}.\, s\ (n\ X\ s\ z))^{\mathsf{N}}}\;\forall\text{-I}
}{(\lambda n^{\mathsf{N}}.\, \Lambda X.\, \lambda s^{X\to X}.\, \lambda z^{X}.\, s\ (n\ X\ s\ z))^{\mathsf{N}\to\mathsf{N}}}\;\to\text{-I}^{n}
$$

Fig. 7. Successor.

The deductive interpretation is the Girard projection of the inductive interpretation.

Proposition 11 (*Deduction Lemma*). *Let* $\mathbf{N}$ *be the inductive interpretation of the naturals, and* $\boldsymbol{s}$ *and* $\boldsymbol{z}$ *be the proofs in the Constructor Lemma; in which* N, s, *and* z *may be taken to be uninterpreted constants. The definitions of* N, s, *and* z *given above may be derived from these by applying the Girard projection.*

$$\mathbf{N}^{\circ} \equiv \mathsf{N} \qquad \left(\frac{\vdots\, s}{n \in \mathbf{N} \to \mathsf{s}\, n \in \mathbf{N}}\right)^{\circ} \equiv \mathsf{s}^{\mathsf{N}\to\mathsf{N}} \qquad \left(\frac{\vdots\, z}{\mathsf{z} \in \mathbf{N}}\right)^{\circ} \equiv \mathsf{z}^{\mathsf{N}}$$

Proof. Straightforward. The Girard projection for zero and successor appears in Figs. 6 and 7. □

In what follows, we assume N, s, and z have their deductive definitions. With this assumption, we will be able to demonstrate a close relation between parametricity and inductivity. We will show that the naturals satisfying parametricity are exactly the same as the naturals satisfying induction,

$$\mathsf{N}^{*\ddagger} = \mathbf{N}^{=}.$$

Equivalently, $\forall n, n'. (n, n') \in \mathsf{N}^{*\ddagger} \leftrightarrow n = n' \wedge n \in \mathbf{N}$.

Reynolds and Plotkin [40] were the first to suggest that parametricity implies inductivity, and Hasegawa [16] was the first to suggest the converse. What we show here is that the values of type N satisfying induction are exactly the same as the values satisfying binary parametricity. This is a stronger result — Reynolds and Plotkin and Hasegawa (and, in earlier work, myself), only consider the case where all values satisfy binary parametricity, and hence N contains exactly the inductive naturals. The result above holds even if some values fail to satisfy binary parametricity. In particular, the result given here applies to models that include $\perp$ as a value of type N, though $\perp$ is not an inductive natural and fails to satisfy parametricity. (The earlier versions of this paper contain essentially the same proofs, but did not extract the stronger conclusion.)

We begin with some useful lemmas. Binary parametricity implies equality, and binary parametricity implies unary parametricity.

Proposition 12 (*Extensive Lemma*). $\forall n, n'. (n, n') \in \mathsf{N}^{*\ddagger} \to n = n'$.

Proof.

$$\begin{array}{ll}
 & (n, n') \in \mathsf{N}^{*\ddagger} \\
\equiv & \quad \text{definition} \\
 & \forall X, X'. \boldsymbol{\forall \mathcal{X}}^{X \times X'}\boldsymbol{.} \forall s, s'. (\forall m, m'. (m, m') \in \boldsymbol{\mathcal{X}} \to (s\ m, s'\ m') \in \boldsymbol{\mathcal{X}}) \to \\
 & \quad \forall z, z'. (z, z') \in \boldsymbol{\mathcal{X}} \to (n\ X\ s\ z, n'\ X'\ s'\ z') \in \boldsymbol{\mathcal{X}} \\
\to & \quad \text{instantiate } \{(n, n') \mid n = n'\}/\boldsymbol{\mathcal{X}} \\
 & \forall X. \forall s, s'. (\forall m, m'. m = m' \to s\ m = s'\ m') \to \forall z, z'. z = z' \to n\ X\ s\ z = n'\ X'\ s'\ z' \\
\to & \quad \text{simplify} \\
 & \forall X. \forall s. \forall z. n\ X\ s\ z = n'\ X\ s\ z \\
\to & \quad \text{extensionality} \\
 & n = n' \quad \square
\end{array}$$

Proposition 13 (*Halving Lemma*). $\forall n, n'. (n, n') \in \mathsf{N}^{*\ddagger} \to n \in \mathsf{N}^{*}$

Proof.

$$
\begin{array}{ll}
 & (n, n') \in \mathsf{N}^{*\ddagger} \\
\equiv & \quad \text{definition} \\
 & \forall X, X'. \boldsymbol{\forall} \boldsymbol{\mathcal{X}}^{X \times X'}\boldsymbol{.}\, \forall s, s'. (\forall m, m'. (m, m') \in \boldsymbol{\mathcal{X}} \rightarrow (s\ m, s'\ m') \in \boldsymbol{\mathcal{X}}) \rightarrow \\
 & \quad \forall z, z'. (z, z') \in \boldsymbol{\mathcal{X}} \rightarrow (n\ X\ s\ z, n'\ X'\ s'\ z') \in \boldsymbol{\mathcal{X}} \\
\rightarrow & \quad \text{instantiate } X/X,\ X'/X',\ \{(x, x') \mid x \in \boldsymbol{\mathcal{X}}^X\}/\boldsymbol{\mathcal{X}}^{X \times X'}, \text{ generalize over } X, \boldsymbol{\mathcal{X}}^X \\
 & \forall X. \boldsymbol{\forall} \boldsymbol{\mathcal{X}}^X\boldsymbol{.}\, \forall s. (\forall m. m \in \boldsymbol{\mathcal{X}} \rightarrow s\ m \in \boldsymbol{\mathcal{X}}) \rightarrow \forall z. z \in \boldsymbol{\mathcal{X}} \rightarrow n\ X\ s\ z \in \boldsymbol{\mathcal{X}} \\
\rightarrow & \quad \text{definition} \\
 & n \in \mathsf{N}^* \quad \square
\end{array}
$$

Böhm and Berarducci [4, Theorem 7.3] observe every term n of type N in F2 satisfies $n\ \mathsf{N}\ \mathsf{s}\ \mathsf{z} = n$, where equality is $\beta\eta$ equality. Here we show a similar result for any terms satisfying binary parametricity.

Proposition 14 (*Böhm and Berarducci's Lemma*). $\forall n. (n, n) \in \mathsf{N}^{*\ddagger} \rightarrow n\ \mathsf{N}\ \mathsf{s}\ \mathsf{z} = n$.

Proof.

$$
\begin{array}{ll}
 & (n, n) \in \mathsf{N}^{*\ddagger} \\
\equiv & \quad \text{definition} \\
 & \forall X, X'. \boldsymbol{\forall} \boldsymbol{\mathcal{X}}^{X \times X'}\boldsymbol{.}\, \forall s, s'. (\forall m, m'. (m, m') \in \boldsymbol{\mathcal{X}} \rightarrow (s\ m, s'\ m') \in \boldsymbol{\mathcal{X}}) \rightarrow \\
 & \quad \forall z, z'. (z, z') \in \boldsymbol{\mathcal{X}} \rightarrow (n\ X\ s\ z, n\ X'\ s'\ z') \in \boldsymbol{\mathcal{X}} \\
\rightarrow & \quad \text{instantiate } \mathsf{N}/X,\ X/X',\ \{(n, x) \mid n\ X\ s\ z = x\}/\boldsymbol{\mathcal{X}},\ \mathsf{s}/s,\ s/s',\ \mathsf{z}/z,\ z/z' \\
 & (\forall m, m'. m\ X\ s\ z = m' \rightarrow (\mathsf{s}\ m)\ X\ s\ z = s\ m') \rightarrow \\
 & \quad \mathsf{z}\ X\ s\ z = z \rightarrow (n\ \mathsf{N}\ \mathsf{s}\ \mathsf{z})\ X\ s\ z = n\ X\ s\ z \\
\rightarrow & \quad \text{simplify} \\
 & (\forall m, m'. (\mathsf{s}\ m)\ X\ s\ z = s\ (m\ X\ s\ z)) \rightarrow \mathsf{z}\ X\ s\ z = z \rightarrow (n\ \mathsf{N}\ \mathsf{s}\ \mathsf{z})\ X\ s\ z = n\ X\ s\ z \\
\rightarrow & \quad \text{definition } \mathsf{s}, \mathsf{z} \\
 & (n\ \mathsf{N}\ \mathsf{s}\ \mathsf{z})\ X\ s\ z = n\ X\ s\ z \\
\rightarrow & \quad \text{extensionality} \\
 & n\ \mathsf{N}\ \mathsf{s}\ \mathsf{z} = n \quad \square
\end{array}
$$

Next, we show that every natural that satisfies induction satisfies unary and binary parametricity. The two proofs are quite similar.

Proposition 15 (*Inductive Implies Deductive*). $\forall n. n \in \mathbf{N} \rightarrow n \in \mathsf{N}^*$.

Proof.

$$
\begin{array}{ll}
 & n \in \mathbf{N} \\
\equiv & \quad \text{definition inductive naturals} \\
 & \boldsymbol{\forall} \boldsymbol{\mathcal{X}}^{\mathsf{N}}\boldsymbol{.}\, (\forall m. m \in \boldsymbol{\mathcal{X}} \rightarrow \mathsf{s}\ m \in \boldsymbol{\mathcal{X}}) \rightarrow \mathsf{z} \in \boldsymbol{\mathcal{X}} \rightarrow n \in \boldsymbol{\mathcal{X}} \\
\rightarrow & \quad \text{instantiate } \mathsf{N}^*/\boldsymbol{\mathcal{X}} \\
 & (\forall m. m \in \mathsf{N}^* \rightarrow \mathsf{s}\ m \in \mathsf{N}^*) \rightarrow \mathsf{z} \in \mathsf{N}^* \rightarrow n \in \mathsf{N}^* \\
\rightarrow & \quad \text{Reynolds embedding applied to } \mathsf{s} \text{ and } \mathsf{z} \\
 & n \in \mathsf{N}^* \quad \square
\end{array}
$$

Proposition 16 (*Inductive Implies Parametric*). $\forall n.\, n \in \mathbf{N} \to (n, n) \in \mathsf{N}^{*\ddagger}$

Proof.

$$
\begin{array}{ll}
 & n \in \mathbf{N} \\
\equiv & \quad \text{definition} \\
 & \boldsymbol{\forall} \boldsymbol{\mathcal{X}}^{\mathsf{N}}\boldsymbol{.}\, (\forall m.\, m \in \boldsymbol{\mathcal{X}} \to \mathsf{s}\, m \in \boldsymbol{\mathcal{X}}) \to \mathsf{z} \in \boldsymbol{\mathcal{X}} \to n \in \boldsymbol{\mathcal{X}} \\
\to & \quad \text{instantiate } \{n \mid (n, n) \in \mathsf{N}^{*\ddagger}\}/\boldsymbol{\mathcal{X}} \\
 & (\forall m.\, (m, m) \in \mathsf{N}^{*\ddagger} \to (\mathsf{s}\, m, \mathsf{s}\, m) \in \mathsf{N}^{*\ddagger}) \to (\mathsf{z}, \mathsf{z}) \in \mathsf{N}^{*\ddagger} \to (n, n) \in \mathsf{N}^{*\ddagger} \\
\to & \quad \text{Abstraction Theorem applied to } \mathsf{s} \text{ and } \mathsf{z} \\
 & (n, n) \in \mathsf{N}^{*\ddagger} \quad \square
\end{array}
$$

Finally, we show that the values satisfying parametricity are the inductive naturals.

Proposition 17 (*Parametric Implies Inductive*). $\forall n.\, (n, n) \in \mathsf{N}^{*\ddagger} \to n \in \mathbf{N}$

Proof.

$$
\begin{array}{ll}
 & (n, n) \in \mathsf{N}^{*\ddagger} \\
\to & \quad \text{Halving lemma} \\
 & n \in \mathsf{N}^{*} \\
\equiv & \quad \text{definition} \\
 & \forall X.\, \boldsymbol{\forall} \boldsymbol{\mathcal{X}}\boldsymbol{.}\, \forall s.\, (\forall m.\, m \in \boldsymbol{\mathcal{X}} \to s\, m \in \boldsymbol{\mathcal{X}}) \to \forall z.\, z \in \boldsymbol{\mathcal{X}} \to n\, X\, s\, z \in \boldsymbol{\mathcal{X}} \\
\to & \quad \text{instantiate } \mathsf{N}/X, \mathbf{N}/\boldsymbol{\mathcal{X}}, \mathsf{s}/s, \mathsf{z}/z \\
 & (\forall m.\, m \in \mathbf{N} \to \mathsf{s}\, m \in \mathbf{N}) \to \mathsf{z} \in \mathbf{N} \to n\, \mathsf{N}\, \mathsf{s}\, \mathsf{z} \in \mathbf{N} \\
\to & \quad \text{Constructor Lemma} \\
 & n\, \mathsf{N}\, \mathsf{s}\, \mathsf{z} \in \mathbf{N} \\
\to & \quad \text{Böhm and Berarducci's Lemma} \\
 & n \in \mathbf{N} \quad \square
\end{array}
$$

Combining the above gives the desired result.

Proposition 18 (*Parametricity is Inductive*).

$$\mathsf{N}^{*\ddagger} = \mathbf{N}^{=}$$

An immediate consequence is that $\mathbf{N} = \mathsf{N}^{*}$ if and only if $\mathsf{N}^{*\ddagger} = \mathsf{N}^{*=}$. Hence, the Girard projection followed by the Reynolds embedding is the identity for the induction over the naturals exactly when binary parametricity is equivalent to unary parametricity.

6. Realizability and Girard's Representation Theorem

Apply the Girard projection followed by the Reynolds embedding to the assertion that a given term is a natural number.

$$
\left(\begin{array}{c} \vdots\ \boldsymbol{m} \\ n \in \mathbf{N} \end{array}\right)^{\circ *} \equiv \left(\begin{array}{c} \vdots \\ \boldsymbol{m}^{\circ \mathsf{N}} \end{array}\right)^{*} \equiv \begin{array}{c} \vdots\ \boldsymbol{m}^{\circ *} \\ \boldsymbol{m}^{\circ} \in \mathsf{N}^{*} \end{array}
$$

In the previous section, we saw that $\mathbf{N} = \mathsf{N}^{*}$ exactly when $\mathsf{N}^{*=} = \mathsf{N}^{*\ddagger}$. In this section, we will show that in the above situation that $\boldsymbol{m}^{\circ} = n$ is provable in P2, further strengthening the sense in which the Girard projection and the Reynolds embedding are inverses. From this result we will derive Girard's Representation Theorem [14,15,22].

Propositions

$$\begin{array}{ll}
(t^C \in \mathcal{A}^C)^{\lhd} & \equiv \{z^A \mid (t^C, z^A) \in \mathcal{A}^{\lhd C \times \mathcal{A}^\circ}\} \\
(A \to B)^{\lhd} & \equiv \{z^{A \to B} \mid \forall x^A.\, x \in A^{\lhd} \to z\, x \in B^{\lhd}\} \\
(\forall \mathcal{X}^C.\, B)^{\lhd} & \equiv \{z^{\forall X.\, B} \mid \forall X.\, \forall \mathcal{X}^{C \times X}.\, z\, X \in B^{\lhd}\} \\
(\forall x^C.\, B)^{\lhd} & \equiv \{z^B \mid \forall x^C.\, z \in B^{\lhd}\} \\
(\forall X.\, B)^{\lhd} & \equiv \{z^B \mid \forall X.\, z \in B^{\lhd}\}
\end{array}$$

Predicates

$$\begin{array}{ll}
(\mathcal{X}^C)^{\lhd} & \equiv \mathcal{X}^{C \times X} \\
(\{x^C \mid A\})^{\lhd} & \equiv \{(x^C, z^{\mathcal{A}^\circ}) \mid z \in A^{\lhd}\}
\end{array}$$

Fig. 8. The realizability interpretation (Part 1).

The key to the proof is a realizability interpretation, similar to those studied by Krivine and Parigot [20] and Takeuti [44]. Krivine and Parigot's interpretation is for a logic in which terms are untyped, and is presented in terms of a particular term model of that logic, but it is the direct inspiration for the translation presented here. Takeuti's interpretation is for a logic essentially the same as the one described here, but the interpretation itself differ in several particulars. As we shall see, the realizability interpretation given here is related to both the Girard projection and the Reynolds embedding.

Recall that the Girard projection takes a proposition A into a type A°, and a proof t into a term t°, such that

$$\left(\begin{array}{c} \vdots\, t \\ A \end{array} \right)^\circ \equiv \begin{array}{c} \vdots \\ t^\circ A^\circ \end{array}.$$

The realizability interpretation takes a proposition A into a predicate $A^{\lhd}$ over terms of type A°, and a proof t into a proof $t^{\lhd}$, such that

$$\left(\begin{array}{c} \vdots\, t \\ A \end{array} \right)^{\lhd} \equiv \begin{array}{c} \vdots\, t^{\lhd} \\ t^\circ \in A^{\lhd} \end{array}.$$

The Realizability interpretation is defined in Figs. 8 and 9. It maps implication into a predicate over terms of function type, and quantification over predicates into a predicate over terms of quantified type. The realizability interpretation also takes a predicate $\mathcal{A}$ over individuals of type C into a predicate $\mathcal{A}^{\lhd}$ over pairs of type $C \times \mathcal{A}^\circ$.

The existence of the realizability interpretation corresponds to Krivine and Parigot's *Conservation Lemma*, and the mapping from proofs to proofs shown in Fig. 9 amounts to a diagrammatic display of their proof of that lemma.

The realizability interpretation preserves substitution for terms and predicates, $(B[t/x])^{\lhd} \equiv B^{\lhd}[t/x]$ and $(B[\mathcal{A}/\mathcal{X}^C])^{\lhd} \equiv B^{\lhd}[\mathcal{A}^{\lhd}/\mathcal{X}^{C \times \mathcal{A}^\circ}]$.

As we have seen, the realizability interpretation is closely related to the Girard projection. Surprisingly, it is also closely related to the Reynolds embedding and doubling.

Proposition 19 (*Realizability and the Reynolds Embedding*). *For all types A,*

$$A^{*\lhd} = A^{*\ddagger}.$$

Equivalently, $\forall z, z'.\, z' \in (z \in A^*)^{\lhd} \equiv (z, z') \in A^{*\ddagger}$.

Proofs

$$\left(\frac{\begin{array}{c}[A]^{x}\\ \vdots\, u\\ B\end{array}}{A \to B}\ \to\text{-I}^{x}\right)^{\lhd} \equiv \frac{\dfrac{\dfrac{\begin{array}{c}[x \in A^{\lhd}]^{x}\\ \vdots\, u^{\lhd}\\ u^{\circ} \in B^{\lhd}\end{array}}{(\lambda x^{A}.\,u^{\circ})\;x \in B^{\lhd}}\;\beta}{x \in A^{\lhd} \to (\lambda x^{A}.\,u^{\circ})\;x \in B^{\lhd}}\;\to\text{-I}^{x}}{\forall x^{A}.\,x \in A^{\lhd} \to (\lambda x^{A}.\,u^{\circ})\;x \in B^{\lhd}}\;\forall^{1}\text{-I}$$

$$\left(\frac{\begin{array}{c}\vdots\, s\\ A \to B\end{array} \qquad \begin{array}{c}\vdots\, t\\ A\end{array}}{B}\ \to\text{-E}\right)^{\lhd} \equiv \frac{\dfrac{\begin{array}{c}\vdots\, s^{\lhd}\\ \forall x^{A}.\,x \in A^{\lhd} \to s^{\circ}\;x \in B^{\lhd}\end{array}}{t^{\circ} \in A^{\lhd} \to s^{\circ}\;t^{\circ} \in B^{\lhd}}\;\forall^{1}\text{-E} \qquad \begin{array}{c}\vdots\, t^{\lhd}\\ t^{\circ} \in A^{\lhd}\end{array}}{s^{\circ}\;t^{\circ} \in B^{\lhd}}\;\to\text{-E}$$

$$\left(\frac{\begin{array}{c}\vdots\, u\\ B\end{array}}{\forall \mathcal{X}^{C}.\,B}\ \forall\text{-I}\right)^{\lhd} \equiv \frac{\dfrac{\dfrac{\begin{array}{c}\vdots\, u^{\lhd}\\ u^{\circ} \in B^{\lhd}\end{array}}{(\Lambda X.\,u^{\circ})\;X \in B^{\lhd}}\;\beta}{\forall \mathcal{X}^{C\times X}.\,(\Lambda X.\,u^{\circ})\;X \in B^{\lhd}}\;\forall\text{-I}}{\forall X.\,\forall \mathcal{X}^{C\times X}.\,(\Lambda X.\,u^{\circ})\;X \in B^{\lhd}}\;\forall^{2}\text{-I}$$

$$\left(\frac{\begin{array}{c}\vdots\, s\\ \forall \mathcal{X}^{C}.\,B\end{array}}{B[\mathcal{A}^{C}/\mathcal{X}]}\ \forall\text{-E}\right)^{\lhd} \equiv \frac{\dfrac{\begin{array}{c}\vdots\, s^{\lhd}\\ \forall X.\,\forall \mathcal{X}^{C\times X}.\,s^{\circ}\;X \in B^{\lhd}\end{array}}{\forall \mathcal{X}^{C\times A}.\,s^{\circ}\;A \in B^{\lhd}[A/X]}\;\forall^{2}\text{-E}}{s^{\circ}\;A \in B^{\lhd}[A/X,\,\mathcal{A}^{\lhd C\times A}/\mathcal{X}]}\;\forall\text{-E}$$

$$\left(\frac{\begin{array}{c}\vdots\, u\\ B\end{array}}{\forall x^{C}.\,B}\ \forall^{1}\text{-I}\right)^{\lhd} \equiv \frac{\begin{array}{c}\vdots\, u^{\lhd}\\ u^{\circ} \in B^{\lhd}\end{array}}{\forall x^{C}.\,u^{\circ} \in B^{\lhd}}\;\forall^{1}\text{-I} \qquad \left(\frac{\begin{array}{c}\vdots\, s\\ \forall x^{C}.\,B\end{array}}{B[t^{C}/x]}\ \forall^{1}\text{-E}\right)^{\lhd} \equiv \frac{\begin{array}{c}\vdots\, s^{\lhd}\\ \forall x^{C}.\,s^{\circ} \in B^{\lhd}\end{array}}{s^{\circ} \in B^{\lhd}[t^{C}/x]}\;\forall^{1}\text{-E}$$

$$\left(\frac{\begin{array}{c}\vdots\, u\\ B\end{array}}{\forall X.\,B}\ \forall^{2}\text{-I}\right)^{\lhd} \equiv \frac{\begin{array}{c}\vdots\, u^{\lhd}\\ u^{\circ} \in B^{\lhd}\end{array}}{\forall X.\,u^{\circ} \in B^{\lhd}}\;\forall^{2}\text{-I} \qquad \left(\frac{\begin{array}{c}\vdots\, s\\ \forall X.\,B\end{array}}{B[A/X]}\ \forall^{2}\text{-E}\right)^{\lhd} \equiv \frac{\begin{array}{c}\vdots\, s^{\lhd}\\ \forall X.\,s^{\circ} \in B^{\lhd}\end{array}}{s^{\circ} \in B^{\lhd}[A/X]}\;\forall^{2}\text{-E}$$

$$\left(\frac{\begin{array}{c}\vdots\, t\\ A\end{array}}{B}\ \beta\right)^{\lhd} \equiv \frac{\begin{array}{c}\vdots\, t^{\lhd}\\ t^{\circ} \in A^{\lhd}\end{array}}{t^{\circ} \in B^{\lhd}}\;\beta$$

Fig. 9. The realizability interpretation (Part 2).

Proof. By induction over the structure of types. Below is the case for $A \to B$, the cases for X and $\forall X.\, B$ are similar.

$$
\begin{array}{ll}
 & z' \in (z \in (A \to B)^*)^{\lhd} \\
\equiv & \quad \text{definition Reynolds embedding} \\
 & z' \in (\forall x.\, x \in A^* \to z\, x \in B^*)^{\lhd} \\
\equiv & \quad \text{definition realizability interpretation} \\
 & \forall x.\, \forall x'.\, x' \in (x \in A^*)^{\lhd} \to z'\, x' \in (z\, x \in B^*)^{\lhd} \\
\equiv & \quad \text{induction hypothesis} \\
 & \forall x.\, \forall x'.\, (x, x') \in A^{*\ddagger} \to (z\, x, z'\, x') \in B^{*\ddagger} \\
\equiv & \quad \text{definition Reynolds embedding, doubling} \\
 & (z, z') \in (A \to B)^{*\ddagger} \quad \square
\end{array}
$$

Combining the above with the results of the previous section, we see that $\mathsf{N}^{*\lhd} = \mathsf{N}^{*\ddagger} = \mathbf{N}^{=}$. Next we give a similar result, where N^* is replaced by $\mathbf{N}$.

Proposition 20 (*Krivine and Parigot's Lemma*).

$$\mathbf{N}^{\lhd} = \mathbf{N}^{=}.$$

Equivalently, $\forall n, n'.\, n' \in (n \in \mathbf{N})^{\lhd} \leftrightarrow n = n' \wedge n \in \mathbf{N}$.

Proof. The proof has three parts.

(i) $n' \in (n \in \mathbf{N})^{\lhd} \to n = n'$. The proof is similar to that of Böhm and Berarducci's Lemma.

$$
\begin{array}{ll}
 & n' \in (n \in \mathbf{N})^{\lhd} \\
\equiv & \quad \text{definition} \\
 & n' \in (\forall \mathcal{X}^{\mathsf{N}}.\, (\forall m.\, m \in \mathcal{X} \to \mathsf{s}\, m \in \mathcal{X}) \to \mathsf{z} \in \mathcal{X} \to n \in \mathcal{X})^{\lhd} \\
\equiv & \quad \text{definition} \\
 & \forall X.\, \forall \mathcal{X}^{\mathsf{N} \times X}.\, \forall s.\, (\forall m.\, \forall m'.\, (m, m') \in \mathcal{X} \to (\mathsf{s}\, m, s\, m') \in \mathcal{X}) \to \forall z.\, (\mathsf{z}, z) \in \mathcal{X} \to (n, n'\, X\, s\, z) \in \mathcal{X} \\
\to & \quad \text{instantiate } X/X,\ \{(n, n') \mid n\, X\, s\, z = n'\}/\mathcal{X},\ s/s,\ z/z \\
 & (\forall m.\, \forall m'.\, m\, s\, z = m' \to (\mathsf{s}\, m)\, X\, s\, z = s\, m') \to \mathsf{z}\, X\, s\, z = z \to n\, X\, s\, z = n'\, X\, s\, z \\
\to & \quad \text{definition } \mathsf{s}, \mathsf{z} \\
 & n\, X\, s\, z = n'\, X\, s\, z \\
\to & \quad \text{extensionality} \\
 & n = n'
\end{array}
$$

(ii) $n' \in (n \in \mathbf{N})^{\lhd} \to n \in \mathbf{N}$. This is straightforward.

$$
\begin{array}{ll}
 & n' \in (n \in \mathbf{N})^{\lhd} \\
\equiv & \quad \text{definition} \\
 & \forall X.\, \forall \mathcal{X}^{\mathsf{N} \times X}.\, \forall s.\, (\forall m.\, \forall m'.\, (m, m') \in \mathcal{X} \to (\mathsf{s}\, m, s\, m') \in \mathcal{X}) \to \forall z.\, (\mathsf{z}, z) \in \mathcal{X} \to (n, n'\, X\, s\, z) \in \mathcal{X} \\
\to & \quad \text{instantiate } X/X,\ \{(n, x) \mid n \in \mathcal{X}\}/\mathcal{X},\ s/s,\ z/z, \text{ generalize on } \mathcal{X} \\
 & \forall \mathcal{X}^{\mathsf{N}}.\, (\forall m.\, m \in \mathcal{X} \to \mathsf{s}\, m \in \mathcal{X}) \to \mathsf{z} \in \mathcal{X} \to n \in \mathcal{X} \\
\equiv & \quad \text{definition} \\
 & n \in \mathbf{N}
\end{array}
$$

(iii) $n \in \mathbf{N} \to n \in (n \in \mathbf{N})^{\lhd}$. We use the fact that $n \in \mathbf{N}$ implies $(n, n) \in \mathsf{N}^{*\ddagger}$. The proof exploits Böhm and Berarducci's Lemma.

$$
\begin{array}{ll}
 & (n, n) \in \mathsf{N}^{*\ddagger} \\
\equiv & \quad \text{definition} \\
 & \forall X, X'.\, \boldsymbol{\forall} \boldsymbol{\mathcal{X}}^{X \times X'}\boldsymbol{.}\, \forall s, s'.\, (\forall m, m'.\, (m, m') \in \boldsymbol{\mathcal{X}} \to (s\; m, s'\; m') \in \boldsymbol{\mathcal{X}}) \to \\
 & \quad \forall z, z'.\, (z, z') \in \boldsymbol{\mathcal{X}} \to (n\; X\; s\; z, n\; X'\; s'\; z') \in \boldsymbol{\mathcal{X}} \\
\to & \quad \text{instantiate } \mathsf{N}/X,\, X/X',\, \mathsf{s}/s,\, s/s',\, \mathsf{z}/z,\, z/z' \\
 & \forall X.\, \boldsymbol{\forall} \boldsymbol{\mathcal{X}}^{\mathsf{N} \times X}\boldsymbol{.}\, \forall s.\, (\forall m, m'.\, (m, m') \in \boldsymbol{\mathcal{X}} \to (\mathsf{s}\; m, s\; m') \in \boldsymbol{\mathcal{X}}) \to \\
 & \quad \forall z.\, (\mathsf{z}, z) \in \boldsymbol{\mathcal{X}} \to (n\; \mathsf{N}\; \mathsf{s}\; \mathsf{z}, n\; X\; s\; z) \in \boldsymbol{\mathcal{X}} \\
\to & \quad \text{Böhm and Berarducci's Lemma} \\
 & \forall X.\, \boldsymbol{\forall} \boldsymbol{\mathcal{X}}^{\mathsf{N} \times X}\boldsymbol{.}\, \forall s.\, (\forall m, m'.\, (m, m') \in \boldsymbol{\mathcal{X}} \to (\mathsf{s}\; m, s\; m') \in \boldsymbol{\mathcal{X}}) \to \\
 & \quad \forall z.\, (\mathsf{z}, z) \in \boldsymbol{\mathcal{X}} \to (n, n\; X\; s\; z) \in \boldsymbol{\mathcal{X}} \\
\equiv & \quad \text{definition} \\
 & n \in (n \in \mathbf{N})^{\lhd} \quad \square
\end{array}
$$

As a corollary, we have that the Girard projection takes a proof that a term belongs to the inductive naturals into *the same* term in the deductive naturals.

Proposition 21 (*Value Representation*). *Let* $\boldsymbol{m}$ *be a proof that some term* n *satisfies* $\mathbf{N}$. *Then*

$$
\left(\begin{array}{c} \vdots\; \boldsymbol{m} \\ n \in \mathbf{N} \end{array} \right)^{\circ} \equiv \begin{array}{c} \vdots \\ \boldsymbol{m}^{\circ\mathbf{N}^{\circ}} \end{array}
$$

and $\boldsymbol{m}^{\circ} = n$ *is provable in* P2.

Proof. The realizability interpretation gives us

$$
\left(\begin{array}{c} \vdots\; \boldsymbol{m} \\ n \in \mathbf{N} \end{array} \right)^{\lhd} \equiv \begin{array}{c} \vdots\; \boldsymbol{m}^{\lhd} \\ \boldsymbol{m}^{\circ} \in (n \in \mathbf{N})^{\lhd} \end{array}
$$

from which Krivine and Parigot's Lemma deduces $\boldsymbol{m}^{\circ} = n$. $\square$

For example, let $\mathsf{2} = \mathsf{s}\;(\mathsf{s}\;\mathsf{z})$, and say we have a proof $\boldsymbol{m}$ in P2 that $\mathsf{2} \in \mathbf{N}$. It is easy to find a proof such that its Girard projection $\boldsymbol{m}^{\circ}$ is the second Church numeral, but can we be sure that this is true for *any* such proof? This is ensured by value representation, which guarantees $\boldsymbol{m}^{\circ} = n$.

A similar result holds for functions. We give the result here for unary functions, but it extends easily to any number of arguments.

Proposition 22 (*Representation Theorem*). *Let* $\boldsymbol{f}$ *be a proof that* $\forall x.\, x \in \mathbf{N} \to g\; x \in \mathbf{N}$, *in which* g *may be an uninterpreted constant known only to satisfy some equations. Then the Girard projection* $\boldsymbol{f}^{\circ}$ *is a term of polymorphic lambda calculus that represents* g, *in that* $\boldsymbol{f}^{\circ}\; n = g\; n$ *for all* $n \in \mathbf{N}$.

Proof. Let $\boldsymbol{m}$ be a proof that $n \in \mathbf{N}$ for some term n. Then

$$
\left(\dfrac{\dfrac{\begin{array}{c} \vdots\; \boldsymbol{f} \\ \forall x.\, x \in \mathbf{N} \to g\; x \in \mathbf{N} \end{array}}{n \in \mathbf{N} \to g\; n \in \mathbf{N}}\,\forall^{1}\text{-E} \qquad \begin{array}{c} \vdots\; \boldsymbol{m} \\ n \in \mathbf{N} \end{array}}{g\; n \in \mathbf{N}}\,\boldsymbol{\to}\textbf{-E} \right)^{\circ} \equiv \dfrac{\begin{array}{c} \vdots \\ \boldsymbol{f}^{\circ\mathsf{N} \to \mathsf{N}} \end{array} \qquad \begin{array}{c} \vdots \\ \boldsymbol{m}^{\circ\mathsf{N}} \end{array}}{(\boldsymbol{f}^{\circ}\; \boldsymbol{m}^{\circ})^{\mathsf{N}}}\,\to\text{-E}
$$

We have $\boldsymbol{m}^{\circ} = n$ and $\boldsymbol{f}^{\circ}\; \boldsymbol{m}^{\circ} = g\; n$ by Krivine and Parigot's lemma. $\square$

$$
\left(
\begin{prooftree}
\AxiomC{$[m \in \mathbf{N}]^{m}$}
\RightLabel{β}
\UnaryInfC{$\forall \mathcal{X}^{\mathsf{N}}.\, (\forall m^{\mathsf{N}}.\, m \in \mathcal{X} \to \mathsf{s}\, m \in \mathcal{X}) \to \mathsf{z} \in \mathcal{X} \to m \in \mathcal{X}$}
\RightLabel{$\forall$-E}
\UnaryInfC{$(\forall m^{\mathsf{N}}.\, m + n \in \mathbf{N} \to (\mathsf{s}\, m) + n \in \mathbf{N}) \to \mathsf{z} + n \in \mathbf{N} \to m + n \in \mathbf{N}$}
\AxiomC{$\vdots\, s$}
\noLine
\UnaryInfC{$\forall n^{\mathsf{N}}.\, n \in \mathbf{N} \to \mathsf{s}\, n \in \mathbf{N}$}
\RightLabel{$\forall^1$-E}
\UnaryInfC{$m + n \in \mathbf{N} \to \mathsf{s}\,(m + n) \in \mathbf{N}$}
\RightLabel{β^+}
\UnaryInfC{$m + n \in \mathbf{N} \to (\mathsf{s}\, m) + n \in \mathbf{N}$}
\RightLabel{$\forall^1$-I}
\UnaryInfC{$\forall m^{\mathsf{N}}.\, m + n \in \mathbf{N} \to (\mathsf{s}\, m) + n \in \mathbf{N}$}
\RightLabel{$\to$-E}
\BinaryInfC{$\mathsf{z} + n \in \mathbf{N} \to m + n \in \mathbf{N}$}
\AxiomC{$[n \in \mathbf{N}]^{n}$}
\RightLabel{β^+}
\UnaryInfC{$\mathsf{z} + n \in \mathbf{N}$}
\RightLabel{$\to$-E}
\BinaryInfC{$m + n \in \mathbf{N}$}
\RightLabel{$\to$-I^{n}}
\UnaryInfC{$n \in \mathbf{N} \to m + n \in \mathbf{N}$}
\RightLabel{$\to$-I^{m}}
\UnaryInfC{$m \in \mathbf{N} \to n \in \mathbf{N} \to m + n \in \mathbf{N}$}
\end{prooftree}
\right)^{\circ}
$$

$$
\equiv
\begin{prooftree}
\AxiomC{$[m^{\mathsf{N}}]$}
\RightLabel{$\forall$-E}
\UnaryInfC{$(m\ \mathsf{N})^{(\mathsf{N}\to\mathsf{N})\to\mathsf{N}\to\mathsf{N}}$}
\AxiomC{$\vdots$}
\noLine
\UnaryInfC{$\mathsf{s}^{\mathsf{N}\to\mathsf{N}}$}
\RightLabel{$\to$-E}
\BinaryInfC{$(m\ \mathsf{N}\ \mathsf{s})^{\mathsf{N}\to\mathsf{N}}$}
\AxiomC{$[n^{\mathsf{N}}]$}
\RightLabel{$\to$-E}
\BinaryInfC{$(m\ \mathsf{N}\ \mathsf{s}\ n)^{\mathsf{N}}$}
\RightLabel{$\to$-I^{n}}
\UnaryInfC{$(\lambda n^{\mathsf{N}}.\, m\ \mathsf{N}\ \mathsf{s}\ n)^{\mathsf{N}\to\mathsf{N}}$}
\RightLabel{$\to$-I^{m}}
\UnaryInfC{$(\lambda m^{\mathsf{N}}.\, \lambda n^{\mathsf{N}}.\, m\ \mathsf{N}\ \mathsf{s}\ n)^{\mathsf{N}\to\mathsf{N}\to\mathsf{N}}$}
\end{prooftree}
$$

Fig. 10. Plus.

For example, let $\mathsf{p}^{\mathsf{N}\to\mathsf{N}\to\mathsf{N}}$ be an uninterpreted symbol satisfying the following equations, where we write $m + n$ for $\mathsf{p}\, m\, n$.

$$\begin{aligned} \mathsf{z} + n &= n \\ (\mathsf{s}\, m) + n &= \mathsf{s}\,(m + n) \end{aligned}$$

Say there is a proof $\boldsymbol{p}$ in P2 that the sum of two naturals is a natural. Applying the Girard projection to that proof yields a term $\boldsymbol{p}^\circ$ in F2 that takes two naturals into a natural.

$$\left(\begin{array}{c} \vdots\ \boldsymbol{p} \\ \forall m, n.\, m \in \mathbf{N} \to n \in \mathbf{N} \to m + n \in \mathbf{N} \end{array}\right)^\circ \equiv \begin{array}{c} \vdots \\ \boldsymbol{p}^{\circ\,\mathsf{N}\to\mathsf{N}\to\mathsf{N}} \end{array}$$

It follows from the Representation Theorem that $\boldsymbol{p}^\circ\, m\, n = m + n$, so from a proof in P2 that sum takes naturals to naturals we have derived a term in F2 that computes sums. An example of such a proof $\boldsymbol{p}$ and the corresponding term $\boldsymbol{p}^\circ$ is displayed in Fig. 10. In this figure, the two rules labelled β^+ are appeals to the equations given above, the proof $\boldsymbol{s}$ the term $\mathsf{s} \equiv \boldsymbol{s}^\circ$ are as in Fig. 7.

This is remarkable. We start with a proof in which p is an uninterpreted symbol. The Girard projection throws away all occurrences of p, indeed it throws away all terms in the proof, as well as all quantifiers over individuals and types. Yet it is guaranteed that the constructed lambda term represents the original function! It almost seems like magic, and, as with the best of magic tricks, knowing how it is done makes it more magical still.

Acknowledgements

My thanks to Rasmus Møgelberg, Martin Odersky, Andrew Pitts, Gordon Plotkin, Jon Riecke, Colin Runciman, Clemens Szyperski, René Vestergaard, to referees of POPL 1999, LICS 2000, TACS 2001, *Information and Computation*, and *Theoretical Computer Science*, and to seminar audiences at JAIST, Bath, and Birmingham. Thanks also to Don Sanella for editing this article, and to the anonymous reviewers.

References

[1] M. Abadi, L. Cardelli, P.-L. Curien, Formal parametric polymorphism, Theoretical Computer Science 121 (1–2) (1993) 9–58. (Part of A Collection of Contributions in Honour of Corrado Boehm on the Occasion of his 70th Birthday.) Also appeared as SRC Research Report 109.
[2] H. Barendregt, Introduction to generalized types systems, Journal of Functional Programming 1 (2) (1991) 125–154.
[3] E.S. Bainbridge, P.J. Freyd, A. Scedrov, P.J. Scott, Functorial polymorphism, in: G. Huet (Ed.), Logical Foundations of Functional Programming, Addison-Wesley, 1990, pp. 315–330.
[4] C. Böhm, A. Berarducci, Automatic synthesis of typed Λ-programs on term algebras, Theoretical Computer Science 39 (2–3) (1985) 135–154.
[5] V. Breazu-Tannen, T. Coquand, Extensional models for polymorphism, Theoretical Computer Science 59 (1988) 85–114.
[6] H.B. Curry, R. Feys, Combinatory Logic, North Holland, 1958.
[7] T. Coquand, G. Huet, The calculus of constructions, Information and Computation 76 (1988) 95–120.
[8] R. Constable, et al., Implementing mathematics with the Nuprl proof development system, Prentice-Hall, 1986.
[9] N.G. de Bruijn, The mathematical language of AUTOMATH, its usage and some of its extensions, in: Proceedings of the Symposium on Automatic Demonstration, in: LNCS, vol. 125, Springer-Verlag, 1970.
[10] L. Damas, R. Milner, Principal type schemes for functional programs, in: 9'th Annual Symposium on Principles of Programming Languages, Albuquerque, N.M., ACM, January 1982.
[11] G. Frege, Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought (1879), in: Jan van Heijenoort (Ed.), From Frege to Gödel: A Source Book in Mathematical Logic, Harvard University Press, 1967, pp. 1879–1931.
[12] P. Freyd, POLYNAT in PER, Contemporary Mathematics 92 (1989) 67–68.
[13] G. Gentzen, Investigations into logical deduction, Mathematische Zeitschrift 39 (1935) 176–210, 405–431. Reprinted in M.E. Szabo, editor, The collected papers of Gerhard Gentzen, North Holland, 1969.
[14] J.-Y. Girard, Interprétation functionelle et élimination des coupures dans l'arithmétique d'ordre supérieure, Ph.D. Thesis, Université Paris VII, 1972.
[15] J.-Y. Girard, Y. Lafont, P. Taylor, Proofs and Types, Cambridge University Press, 1989.
[16] R. Hasegawa, Categorical data types in parametric polymorphism, Mathematical Structures in Computer Science 4 (1994) 71–109.
[17] R. Hindley, The principal type scheme of an object in combinatory logic, Transactions of the American Mathematical Society 146 (1969) 29–60.
[18] W.A. Howard, The formulae-as-types notion of construction, in: J.P. Seldin, J.R. Hindley (Eds.), To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism, Academic Press, 1980 (The original version was circulated privately in 1969).
[19] J.M.E. Hyland, E.P. Robinson, G. Rosolini, Algebraic types in PER models, in: Mathematical Foundations of Programming Semantics, New Orleans, in: LNCS, vol. 442, Springer-Verlag, 1989, pp. 333–350.

[20] J.-L. Krivine, M. Parigot, Programming with proofs, Journal of Information Processing and Cybernetics (EIK) 26 (3) (1990) 149–167. Revised version of a lecture presented at the 6'th International Symposium on Computation Theory, SCT '87, Wednisch-Rietz, GDR, 30 November–4 December 1987.

[21] D. Leivant, Reasoning about functional programs and complexity classes associated with type disciplines, in: 24'th Symposium on Foundations of Computer Science, IEEE, Washington, DC, 1983, pp. 460–469.

[22] D. Leivant, Contracting proofs to programs, in: P. Odifreddi (Ed.), Logic and Computer Science, Academic Press, 1990.

[23] H. Mairson, Outline of a proof theory of parametricity, in: J. Hughes (Ed.), 5'th International Conference on Functional Programming Languages and Computer Architecture, Cambridge, Massachusetts, in: LNCS, vol. 523, Springer-Verlag, August 1991.

[24] P. Martin-Löf, Constructive mathematics and computer programming, in: 6'th International Congress for Logic, Methodology, and Philosophy of Science, North Holland, 1982, pp. 153–175.

[25] P. Martin-Löf, A construction of the provable well-orderings of a theory of species, in: A. Anderson, M. Zeleny (Eds.), Logic, Meaning and Computation: Essays in Memory of Alonzo Church, Kluwer–Academic, 2001.

[26] R. Milner, A theory of type polymorphism in programming, Journal of Computers and Systems Science 17 (1978) 348–375.

[27] E. Moggi, Communication to the Types electronic forum, 10 February 1986.

[28] Rasmus Møgelberg, Private communication, March 2004.

[29] Q. Ma, J.C. Reynolds, Types, abstraction, and parametric polymorphism, part 2, in: S. Brookes, et al. (Eds.), Mathematical Foundations of Programming Semantics, in: LNCS, Springer-Verlag, 1991.

[30] A.M. Pitts, Polymorphism Is Set Theoretic, Constructively, in: D.H. Pitt, A. Poigné, D.E. Rydeheard (Eds.), Category Theory and Computer Science, Edinburgh, 1987 pp. 12–39.

[31] A.M. Pitts, Non-trivial power types can't be subtypes of polymorphic types, in: 4th Annual Symposium on Logic in Computer Science, IEEE Computer Society Press, Washington, 1989, pp. 6–13.

[32] A.M. Pitts, Parametric polymorphism and operational equivalence, Technical report 453, Cambridge University Computer Laboratory, 1998.

[33] G. Plotkin, M. Abadi, A logic for parametric polymorphism, in: M. Bezem, J.F. Groote (Eds.), Typed Lambda Calculi and Applications, in: LNCS, vol. 664, Springer-Verlag, March 1993, pp. 361–375.

[34] G. Plotkin, M. Abadi, L. Cardelli, Subtyping and parametricity, in: 9'th Annual Symposium on Logic in Computer Science, July 1994, pp. 310–319.

[35] J.C. Reynolds, Towards a theory of type structure, in: B. Robinet (Ed.), Colloque sur la Programmation, in: LNCS, vol. 19, Springer-Verlag, 1974.

[36] J.C. Reynolds, Types, abstraction, and parametric polymorphism, in: R.E.A. Mason (Ed.), Information Processing, vol. 83, North-Holland, Amsterdam, 1983, pp. 513–523.

[37] J.C. Reynolds, Polymorphism is not set theoretic, in: G. Kahn, D. MacQueen, G. Plotkin (Eds.), Semantics of Data Types, Sophia-Antipolis, France, in: LNCS, vol. 173, Springer-Verlag, 1984, pp. 145–156.

[38] J.C. Reynolds, Introduction to part II: Polymorphic lambda calculus, in: G. Huet (Ed.), Logical Foundations of Functional Programming, Addison-Wesley, 1990.

[39] J.C. Reynolds, Three approaches to type structure, in: Mathematical Foundations of Software Development, in: LNCS, vol. 185, Springer-Verlag, 1985.

[40] J.C. Reynolds, G.D. Plotkin, On Functors Expressible in the polymorphic typed lambda calculus, in: G. Huet (Ed.), Logical Foundations of Functional Programming, Addison-Wesley, 1990, pp. 127–152.

[41] E.P. Robinson, G. Rosolini, Reflexive graphs and parametric polymorphism, in: S. Abramsky (Ed.), 9th Symposium in Logic in Computer Science, IEEE Computer Society, 1994.

[42] I. Rummelhoff, Polynat is not universal in all PER-models. Manuscript, University of Oslo, September 2003.

[43] C. Strachey, Fundamental concepts in programming languages, in: Lecture notes, International Summer School in Computer Programming, Copenhagen, August 1967. Reprinted in Higher-Order and Symbolic Computation 13 (1–2) (2000) 11–49.

[44] I. Takeuti, An axiomatic system of parametricity, Fundamenta Informaticae 33 (1998) 397–432. IOS Press.

[45] P. Wadler, Theorems for free!, in: 4'th International Conference on Functional Programming Languages and Computer Architecture, ACM Press, London, September 1989.

[46] P. Wadler, Recursive types for free!, manuscript, 1991.

[47] P. Wadler, The Girard–Reynolds isomorphism, in: N. Kobayashi, B. Pierce (Eds.), Theoretical Aspects of Computer Software, in: LNCS, vol. 2215, Springer-Verlag, Sendai, Japan, October 2001, pp. 468–491.

[48] P. Wadler, The Girard–Reynolds isomorphism, Information and Computation 186 (2) (2003) 260–284. Special issue of best papers from TACS 2001.