The Girard-Reynolds Isomorphism[1]

Philip Wadler
Avaya Labs
wadler@avaya.com

Version: August 2002

The second-order polymorphic lambda calculus, F2, was independently discovered by Girard and Reynolds. 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*: for a suitable notion of logical relation, every term in F2 takes related arguments into related results. 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. The Girard projection discards all first-order quantifiers, so it seems unreasonable to expect that the Girard projection followed by the Reynolds embedding should also be the identity. However, we show that in the presence of Reynolds's *parametricity* property that this is indeed the case, for propositions corresponding to inductive definitions of naturals or other algebraic types.

1. INTRODUCTION

Double-barrelled names in science may be special for two reasons: some belong to ideas so subtle that they required two collaborators to develop; and some belong to ideas so sublime that they possess two independent discoverers. The Curry-Howard isomorphism is an idea of the first sort that guarantees the existence of ideas of the second sort, such as the Hindley-Milner type system and the Girard-Reynolds polymorphic lambda calculus.

The Curry-Howard isomorphism consists of a correspondence between a logical calculus and a computational calculus. Each logical formula corresponds to a computational type, each logical proof corresponds to a computational term, and reduction of proofs corresponds to reductions of terms. This last point means that it is not just formulas and proofs that are preserved by the correspondence, but the structure between them as well; hence we have no mere bijection but a true isomorphism.

Curry formulated this principle for combinatory logic and combinator terms [CF58], and Howard observed that it also applies to intuitionistic propositional logic and simply-typed lambda terms [How80]. The same idea extends to a correspondence between first-order intuitionistic logic with propositional variables and simply-typed lambda calculus with type variables, which explains why the logician Hindley and the computer scientist Milner independently discovered the Hindley-Milner type system [Hin69, Mil78, DM82]. It also extends to a correspondence

[1] *Information and Computation*, 186(2):260–284, November 2003. (Invited submission, best papers from *Theoretical Aspects of Computer Software*, Sendai, October 2001.)

between second-order intuitionistic logic with quantifiers over proposition variables and second-order typed lambda calculus with quantifiers over type variables, which explains why the logician Girard and the computer scientist Reynolds independently discovered the polymorphic lambda calculus [Gir72, Rey74].

Girard and Reynolds each made additional discoveries about the calculus that bears their name, henceforth 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 (with both first- and second-order quantifiers) can be represented in F2 (using second-order quantifiers only). Reynolds proved an *Abstraction Theorem*: for a suitable notion of logical relation, every term in F2 takes related arguments into related results [Rey83].

The calculus P2 is larger than the image under the Curry-Howard isomorphism of F2: the former has first-order terms (we will take these to be terms of untyped lambda calculus) and both first- and second-order quantifiers, while the latter has second-order quantifiers only. 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 formulas to types and proofs to terms, but differs in that it erases all information about first-order terms and first-order quantifiers. This mapping also preserves reductions, so it is no mere surjection but a true homomorphism.

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 formulas and proofs to terms, but differs in that it adds information about first-order quantifiers and first-order terms. This mapping also preserves reductions, so it is no mere injection but a true homomorphism.

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.

Strachey distinguished two types of polymorphism, where the meaning of a term depends upon a type [Str67]. 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. One consequence of the parametricity condition is the Identity Extension Lemma, which asserts that the relation corresponding to a type is the identity relation, so long as the relation corresponding to any free type variable is also taken to be the identity relation.

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 this duality, its precise description seems to have been more elusive.

Going the other way, it seems unreasonable to expect that the Girard projection followed by the Reynolds embedding should also yield the identity, because the projection discards all information about first-order terms. For instance, here is the standard inductive definition of the naturals in P2.

$$\mathsf{N} \equiv \{\, n \mid \forall X.\, (\forall m.\, m \in X \to \mathsf{s}\, m \in X) \to \mathsf{0} \in X \to n \in X \,\}$$

Here s and $\mathsf{0}$ are the usual successor and zero operations on Church numerals in

untyped lambda calculus. Applying the Girard projection yields, the type of the Church numerals in F2.

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

Then applying the Reynolds embedding in turn yields the following predicate, back in P2.

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

This predicate does not look much like N — it makes no mention of s or $\mathsf{0}$. However we will see that if we assume that the type $\mathbf{N}$ satisfies an analogue of Reynolds's *parametricity* condition, then N and $\mathbf{N}^*$ are indeed equivalent in P2.

Hence, in the presence of parametricity, not only does the Girard projection take N to $\mathbf{N}$, but also the Reynolds embedding takes $\mathbf{N}$ to N, and so in this important case one has not merely an embedding-projection pair but a true isomorphism.

To show these results, we tease apart the Identity Extension Lemma, breaking it into two properties. A type is *extensional* if values which satisfy the logical relation at that type are equal, and a type is *parametric* if values of the type which are equal satisfy the logical relation. It is shown that the naturals are extensional, and that the naturals satisfy induction if and only if they are parametric.

The natural numbers are a special case of an algebraic data type. The representation of algebraic data types in polymorphic lambda calculus was first proposed by Böhm and Berarducci [BB85], 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 [Lei83, Lei90] and Krivine and Parigot [KP90].

This paper contains results only for the naturals, but it appears these results extend straightforwardly to any algebraic type represented as a data system, following any of the works cited above [BB85, Lei83, Lei90, KP90]. (An earlier version of this paper [Wad01] treated algebraic data types using sums, products, and fixpoints rather than data systems. While the treatment of sums and products in that formulation is correct, the treatment of fixpoints may need some work.)

This paper also contains a proof of Reynolds's Abstraction Theorem, and proves Girard's Representation Theorem using a notion of realizability due to Krivine and Parigot [KP90]. These are not so much new proofs, as old proofs clarified. In particular, we set Girard's and Reynolds's proofs in a common framework, highlighting the relationship between them. Unlike some previous work, no sophisticated semantic formalism or specialized logic is required; all is formulated within the well-known system of second-order predicate logic.

Both Girard's and Reynolds's results have spawned large bodies of related work. Tutorials on Girard's Representation Theorem have been written by Girard, Taylor, and Lafont [GLT89] and Leivant [Lei90]. Girard's Representation Theorem has been further explored by Leivant [Lei83, Lei90] and by Krivine and Parigot [KP90], among others. Reynolds's parametricity has been further explored by Reynolds [Rey84, Rey90, MR91], Reynolds and Plotkin [RP90], Bainbridge, Freyd, Scedrov and Scott [BFSS90], Hasegawa [Has94], Pitts [Pit87, Pit89, Pit98], and Wadler [Wad89, Wad91], among others. Formulations of the Abstraction Theorem in terms of logics have been examined by Mairson [Mai91], in various combinations by Abadi, Cardelli, Curien, and Plotkin [ACC93, PA93, PAC94], and by Takeuti [Tak98]. Many of these works observe some connection between parametricity and algebraic

types [RP90, Has94, BFSS90, ACC93, PA93, PAC94, Tak98]. Breazu-Tannen and Coquand [BC88], building on work of Moggi [Mog86], show how to turn any model of untyped lambda calculus into a model of polymorphic lambda calculus that satisfies a parametricity condition at all algebraic types.

In addition to the work of Girard and Reynolds, particularly strong influences on this work include: Böhm and Berarducci [BB85], who first showed how to represent algebraic types in polymorphic lambda calculus; Leivant [Lei90], who presents Girard's result as a projection from P2 to F2; Mairson [Mai91], who presents Reynolds's result as an embedding of F2 in P2; Plotkin and Abadi [PA93] and Takeuti [Tak98], who present typed analogues of the untyped logic used here; and Krivine and Parigot [KP90], who use a logic over untyped lambda terms similar to P2, and present a realizability result similar to the one given here.

The basic structure of the proofs in Section 5 was suggested, independently, by Hasegawa [Has94] and Wadler [Wad91]. Hasegawa [Has94] was also the first to suggest a relation between parametricity and induction. Wadler's proof [Wad91] was not published, but it circulated informally, and influenced the work of Abadi, Cardelli, and Curien [ACC93] and the subsequent work of Plotkin and Abadi [PA93].

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. However, lack of types severely restricts the available models. A logic with typed terms, such as that considered by Plotkin and Abadi [PA93] or Takeuti [Tak98], allows a fuller range of models. Fortunately, it appears straightforward to transpose the results of this paper to such typed logics.

Mairson [Mai91] 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" [Mai91], 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 [deB70], Martin-Löf's type theory [Mar82], Constable's Nuprl [Con86], Coquand and Huet's calculus of constructions [CH88], and Barendregt's lambda cube [Bar91]. Each of these calculi introduces dependent types (types that depend upon values) to map first-order quantifiers into the type system. In contrast, the Girard projection discards all first-order information. To quote Leivant [Lei90],

> 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 much of what matters in computing: the naturals and other algebraic types.

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 Reynolds embedding and the Girard projection, and observes that the embedding followed by the projection is the identity. Section 4 explains doubling and parametricity. Section 5 shows that the two definitions of the naturals are equivalent under the parametricity postulate, and similarly for other algebraic types, and hence that there is a sense in which the Girard projection followed by the Reynolds embedding is also the identity. Section 6 applies a realizability interpretation to prove Girard's Representation Theorem.

2. SECOND-ORDER LAMBDA CALCULUS AND LOGIC

The second-order lambda calculus F2 is summarized in Figure 1, and second-order intuitionistic logic P2 is summarized in Figure 2. In what follows, we use different terminology to distinguish between the two systems, using *typing* and *derivation* for F2 in contrast to *judgement* and *proof* for P2.

For each calculus we list the syntactic categories, the derivation or proof rules, and the reductions that act upon derivations or proofs. We use x, y, z to range over individual variables in both calculi, and use X, Y, Z to range over type variables in F2 and predicate variables in P2. We write $\equiv$ for syntactic equivalence of terms, formulas, derivations, or proofs. In Figures 1 and 2 we write $\Rightarrow$ for a single reduction step between derivations or proofs; and in what follows we also write $\Rightarrow$ for zero or more reduction steps.

Typings in F2 have the form $\Gamma \vdash t : A$, expressing that term t has type A in context Γ, where a context consists of pairs $x : A$ associating individual variables with types. 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, abstraction $\lambda x : A.\, u$, application $s\, t$, type abstraction $\Lambda X.\, u$, and type application $s\, A$. A derivation δ uniquely determines its concluding judgement $\Gamma \vdash t : A$, and conversely, δ is uniquely determined by Γ and t.

A derivation reduces when an introducer is followed by the corresponding eliminator. We write $\delta[x := \epsilon]$ for the proof that results by substituting proof ϵ for each use of (Id) on x in the proof δ. (A technicality: because the place where ϵ is substituted may have additional hypotheses in scope, substitution may require adding redundant hypotheses to the proof of ϵ.)

Judgements in P2 have the form $\Theta \vdash \phi$, expressing that the formula list Θ has the formula ϕ as a consequence. An atomic formula has the form $M \in X$, where M is a term and X is a predicate variable. Formulas are formed from atomic formulas, implication $\phi \to \psi$, quantification over individual variables $\forall x.\, \psi$, and quantification over predicate variables $\forall X.\, \psi$. A term is an untyped lambda term. We write $M =_\beta N$ if M and N can be shown equivalent by β reduction.

Predicate variables X range over properties of terms. Notationally, we treat these as sets. Thus we write $M \in X$ to mean that term M satisfies predicate X. The comprehension notation $\{\, x \mid \phi \,\}$ denotes the predicate of x that is satisfied when the formula ϕ over x holds; so x is free in ϕ but bound in $\{\, x \mid \phi \,\}$. We write $\psi[X := \{\, x \mid \phi \,\}]$ for the formula that results by replacing each occurrence of an atomic formula $M \in X$ in ψ by the formula $\phi[x := M]$. (Some formulations write $X(M)$ or $X\, M$ instead of $M \in X$, and $(x)\phi$ or $x.\phi$ instead of $\{\, x \mid \phi \,\}$.)

A proof reduces when an introducer is followed by the corresponding eliminator. We write $\pi[\phi := \rho]$ for the proof that results by substituting ρ for each use of (Id) on ϕ in the proof of π. (Again, a similar technicality applies.)

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

$$\begin{array}{lcl} \top & \equiv & \forall X.\, () \in X \to () \in X \\ \bot & \equiv & \forall X.\, () \in X \\ \phi \wedge \psi & \equiv & \forall X.\, (\phi \to \psi \to () \in X) \to () \in X \\ \phi \vee \psi & \equiv & \forall X.\, (\phi \to () \in X) \to (\psi \to () \in X) \to () \in X \end{array}$$

Here we assume X does not appear free in ϕ or ψ, and write $()$ as shorthand for $\lambda x.\, x$. (It doesn't matter which term is chosen, any closed term works as well.)

Let $P \equiv \{\, x \mid \phi \,\}$ and $Q \equiv \{\, x \mid \psi \,\}$ be two predicates. We write $P \subseteq Q$ if $\forall x.\, x \in P \to x \in Q$, and we write $P = Q$ if $P \subseteq Q$ and $Q \subseteq P$.

Remarkably, P2 is powerful enough to express equality between terms. Following Leibniz, two terms are equal if one may be substituted for the other. Hence we define,

$$M = N \equiv \forall X.\, M \in X \to N \in X.$$

That is, terms M and N are equal if any property X that holds of M also holds of N. It is easy to see that equality is reflexive.

$$\begin{array}{lcl} & & M = M \\ & \equiv & \quad \text{(definition)} \\ & & \forall X.\, M \in X \to M \in X \end{array}$$

It is more subtle to see that it is symmetric.

$$\begin{array}{lcl} & & M = N \\ & \equiv & \quad \text{(definition)} \\ & & \forall X.\, M \in X \to N \in X \\ & \Rightarrow & \quad \text{(instantiate } X := \{\, x \mid x = M \,\}) \\ & & M = M \to N = M \\ & \Rightarrow & \quad \text{(equality is reflexive)} \\ & & N = M \end{array}$$

One may similarly show transitivity, and that $M =_\beta N$ implies $M = N$.

A logic is *extensional* if whenever two terms are applied to equal arguments they return equal results. We define

$$(\forall x.\, M\ x = N\ x) \to M = N \qquad \text{(ext)}$$

where x does not appear free in M and N. We write P2 + ext for P2 with (ext) as an axiom. Extensionality for untyped terms is a stronger assumption than extensionality for typed terms. Arguably, it is stronger than one might wish, but several of our key proofs depend upon it.

As pointed out by Krivine and Parigot [KP90], the term model with open terms quotiented by β and η gives a model of P2 + ext. (A universe of closed lambda terms causes problems, particularly for η, as discussed by Barendregt [Bar80, Section 17.3].)

3. THE REYNOLDS EMBEDDING AND THE GIRARD PROJECTION

The Reynolds embedding takes a derivation δ of a typing in F2,

$$x_1 : A_1, \ldots, x_n : A_n \vdash u : B,$$

into a proof δ^* of a judgement in P2,

$$x_1 \in A_1^*, \ldots, x_n \in A_n^* \vdash |u| \in B^*.$$

Each type A maps into a predicate A^*, each typed lambda term t maps via type erasure into an untyped lambda term $|t|$, and each typing $t : A$ maps into the formula $|t| \in A^*$. The Reynolds embedding is defined in Figure 3.

The Girard projection takes a proof π of a judgement in P2,

$$\phi_1, \ldots, \phi_n \vdash \psi,$$

into a derivation π° of a typing in F2,

$$x_1 : \phi_1^\circ, \ldots, x_n : \phi_n^\circ \vdash u : \psi^\circ.$$

Here u is a typed term with free variables $x_1, \ldots, x_n$, which is uniquely determined by the proof π. The variables are assumed to be taken from a fixed list $x_1, x_2, \ldots$ of variables. Each formula ϕ maps into a type ϕ°. The Girard projection is defined in Figure 4.

We extend the Girard projection to predicates in the obvious way, taking $(\{\, x \mid \phi \,\})^\circ \equiv \phi^\circ$. Both the Reynolds embedding and the Girard projection preserve substitution, so that

$$(A[X := B])^* \equiv A^*[X := B^*] \quad \text{and} \quad \phi[X := \{\, x \mid \psi \,\}]^\circ \equiv \phi^\circ[X := \psi^\circ].$$

The Girard embedding is a homomorphism, in that it preserves reductions.

PROPOSITION 3.1. *(Girard preserve reductions) If $\pi \Rightarrow \rho$ in* P2*, then $\pi^\circ \Rightarrow \rho^\circ$ in* F2*.*

Proof. It is easy to confirm that the Girard projection takes the $\rightarrow$ and $\forall^2$ reductions in P2 into the corresponding reductions of F2, and the $\forall^1$ reduction and commuting conversions of P2 into the identity. ∎

(An earlier version of this paper [Wad01] claimed that the Reynolds embedding also preserves reductions. This is false, since if one derivation reduces to another then they end in different terms, while if one proof reduces to another then they end in the same formula.)

(Leivant [Lei90] claims that the Girard projection also reflects reductions, that is, that if $\pi^\circ \Rightarrow \epsilon$ then there exists a proof ρ such that $\pi \Rightarrow \rho$ and $\epsilon \equiv \rho^\circ$. This appears false, since (for instance) corresponding introduction and elimination rules may be separated by a $\forall^1$ rule in a proof, but will be adjacent in the corresponding Girard projection. Reflection may hold in the presence of suitable commuting conversions, but Leivant does not mention these.)

We consider judgements and derivations in F2 as equivalent up to renaming of variables. That is, the judgements $x_1 : A_1, \ldots, x_n : A_n \vdash t : B$ and $y_1 : A_1, \ldots, y_n : A_n \vdash u : B$ are equivalent if $t = u[y_1 := x_1, \ldots, y_n := x_n]$.

The Reynolds embedding followed by the Girard projection is the identity.

PROPOSITION 3.2. *(Girard inverts Reynolds) If δ is a derivation in* F2*, then $\delta^{*\circ} \equiv \delta$.*

Proof. Straightforward induction over the structure of derivations. ∎

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

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

Applying the Reynolds embedding yields the following predicate in P2,

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

It is easy to check that $\mathbf{N} \equiv \mathbf{N}^{*\circ}$.

Let $\mathbf{2}$ be the second Church numeral in F2, and let 2 be its erasure, $2 \equiv |\mathbf{2}|$.

$$\begin{array}{lcl} \mathbf{2} & \equiv & \Lambda X.\, \lambda s : X \to X.\, \lambda z : X.\, s\, (s\, z) \\ 2 & \equiv & \lambda s.\, \lambda z.\, s\, (s\, z) \end{array}$$

If δ is the derivation of $\vdash \mathbf{2} : \mathbf{N}$ in F2, then δ^* is the proof of $\vdash 2 \in \mathbf{N}^*$ in P2. It is easy to check that $\delta \equiv \delta^{*\circ}$.

Note that the Girard projection takes equality $M = N$ into the unit type $\forall X.\, X \to X$. In the term model, or in any parametric model, the only value of this type is the identity function [Wad89]. Hence, the Girard projection erases any information content in the proof of an equality judgement.

Also observe that the erasure of the extensionality axiom (ext) has the type $(\forall X.\, X \to X) \to (\forall X.\, X \to X)$. One may extend the Girard projection so that it maps axiom (ext) into the derivation of the term $\lambda i : \forall X.\, X \to X.\, i$.

4. DOUBLING AND PARAMETRICITY

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.

We represent binary relations as predicates over pairs. We use the usual encoding of pairs in lambda calculus, taking $(M, N) \equiv \lambda k.\, kMN$, and $\mathsf{fst} \equiv \lambda z.\, z(\lambda x.\, \lambda y.\, x)$, and $\mathsf{snd} \equiv \lambda z.\, z(\lambda x.\, \lambda y.\, y)$. We also write $\{\, (x, y) \mid \phi \,\}$ as an abbreviation for $\{\, z \mid \phi[x := \mathsf{fst}\ z,\ y := \mathsf{snd}\ z] \,\}$. Observe that $(M, N) \in \{\, (x, y) \mid \phi \,\}$ simplifies to $\phi[x := M,\ y := N]$ as required.

Doubling is defined with the aid of operations that rename variables. For each individual variable x there exists a renaming x'. We write M' for the term that results by renaming each free variable x in M to x'. For each propositional variable X acting on individuals (e.g., $x \in X$) there exists a renaming $\hat{X}$ acting on pairs (e.g., $(x, x') \in \hat{X}$).

Doubling takes a proof π of a judgement in P2,

$$\Theta \vdash \phi,$$

into a proof $\pi^\ddagger$ of a judgement in P2,

$$\Theta^\ddagger \vdash \phi^\ddagger.$$

Each predicate ϕ maps into a predicate $\phi^\ddagger$. Doubling is defined in Figure 5.

We extend doubling to predicates in the obvious way, taking $(\{\, x \mid \phi \,\})^\ddagger \equiv \{\, (x, x') \mid \phi^\ddagger \,\}$. Doubling preserves substitution, so that

$$\phi[X := \psi]^\ddagger \equiv \phi^\ddagger[\hat{X} := \psi^\ddagger].$$

What Reynolds calls the Abstraction Theorem [Rey83] and what Plotkin and Abadi call the Logical Relations Lemma [PA93] arises as the composition of the Reynolds embedding with doubling.

PROPOSITION 4.1. *(Abstraction Theorem) If the following typing is derivable in* F2*,*

$$x_1 : A_1, \ldots, x_n : A_n \vdash t : B,$$

then the following judgement is provable in P2*,*

$$(x_1, x_1') \in A_1^{*\ddagger}, \ldots, (x_n, x_n') \in A_n^{*\ddagger} \vdash (|t|, |t|') \in B^{*\ddagger}.$$

Proof. Immediate. The Reynolds embedding followed by doubling takes a derivation of the former into a proof of the latter. ∎

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

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

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

$$\begin{aligned} \mathbf{N}^{*\ddagger} \quad \equiv \quad & \{\, (n, n') \mid \forall \hat{X}.\, \forall s, s'.\, (\forall m, m'.\, (m, m') \in \hat{X} \to (s\, m, s'\, m') \in \hat{X}) \\ & \qquad \to \forall z, z'.\, (z, z') \in \hat{X} \to (n\, s\, z, n'\, s'\, z') \in \hat{X} \,\} \end{aligned}$$

Here we write $\forall x, x'.\, \phi$ to abbreviate $\forall x.\, \forall x'.\, \phi$.

Similarly, starting with the derivation δ of the second Church numeral $\mathbf{2}$ in F2 and applying the Reynolds embedding followed by doubling yields the derivation $\delta^{*\ddagger}$ in P2 of the judgement

$$\vdash (2, 2) \in \mathbf{N}^{*\ddagger}.$$

Here renaming has no effect on 2, which contain only bound variables.

The identity relation on a type is equality restricted to that type.

DEFINITION 4.2. The *identity relation* on type A is written $A^{=}$.

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

It is easy to verify that $A^{=}$ is symmetric and transitive, that $x \in A^*$ implies $(x, x) \in A^{=}$, and that $(x, x') \in A^{=}$ implies $x \in A^*$ and $x' \in A^*$.

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.

DEFINITION 4.3. The *parametric closure* on type A is written $A^{\approx}$.

$$A^{\approx} \equiv A^{*\ddagger}[\hat{X}_1 := X_1^{=}, \ldots, \hat{X}_n := X_n^{=}]$$

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

It is easy to verify that $A^{\approx}$ is symmetric and transitive, so it is a partial equivalence relation, and that $(x, x') \in A^{\approx}$ implies that $x \in A^*$ and $x' \in A^*$.

It is interesting to consider those cases where the identity relation implies the parametric closure, or conversely.

DEFINITION 4.4. We say that type A is *parametric* when $A^{=} \subseteq A^{\approx}$, and *extensive* when $A^{\approx} \subseteq A^{=}$.

An assertion that every type is parametric corresponds to Reynolds's parametricity condition, and an assertion that every type is parametric and extensive might correspond to Reynolds's Identity Extension Lemma [Rey83].

In Section 5, we will see that the naturals are extensive, and that the naturals are parametric if and only if induction works for the Church numerals.

That doubling takes proofs into proofs suggests that it should be consistent to assume that every type is parametric, since if $M \in A^*$ is provable for any closed term M and closed type A then $(M, M) \in A^{\approx}$ is also provable. However, we do not pursue this point further in this paper.

However, it would not be consistent to assume that every type is both parametric and extensive. Assume that types A and B are parametric and extensive and that $f, g \in (A \to B)^*$. If $A \to B$ is extensive, it follows that $f\, x = g\, x$ for all $x \in A^*$ implies $f = g$. But this is not appropriate, since in the untyped world we may apply f and g to arguments that are not of type A.

It is easy to construct a counter-example demonstrating this problem. (Easy, but not obvious; I'm grateful to an anonymous referee for the following.) As usual, let $\mathsf{0} = \lambda s.\, \lambda z.\, z$, $\mathsf{I} = \lambda x.\, x$, $\mathsf{K} = \lambda x.\, \lambda y.\, x$, then take $f = \lambda n.\, \mathsf{0}$ and $g = \lambda n.\, n\, \mathsf{I}\, \mathsf{0}$. We have $f\, n = \mathsf{0} = g\, n$ for every $n \in \mathbf{N}^*$. (The second equality is easily proved by induction, which is justified by the results in Section 5 on the assumption that $\mathbf{N}$ is parametric.) Clearly, $f, g \in (\mathbf{N} \to \mathbf{N})^*$, so if $\mathbf{N} \to \mathbf{N}$ is extensive we would conclude $f = g$. But this is false, since $f\, \mathsf{K} = \mathsf{0} \neq \mathsf{I} = g\, \mathsf{K}$.

These considerations suggest that we do not get an analogue of the Identity Extension Lemma: we cannot in general expect $(x, x') \in A^{\approx}$ to be equivalent to $x = x'$ and $x \in A^*$. For closed types it may be appropriate to simply take $A^{\approx}$ as the definition of equality at type A, in which case the Identity Extension Lemma holds by definition, though for open types the story is less clear.

5. PARAMETRICITY AND INDUCTIVITY

We consider two definitions of the natural numbers, an *inductive* definition, N, and a *deductive* definition $\mathbf{N}$.

The inductive definition N corresponds to the induction principle for natural numbers.

$$\mathsf{N} \equiv \{\, n \mid \forall X.\, (\forall m.\, m \in X \to \mathsf{s}\, m \in X) \to \mathsf{0} \in X \to n \in X \,\}$$

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 $\mathsf{0}$ 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 [Fre79].

One immediate consequence of the definition is that s and $\mathsf{0}$ do indeed construct natural numbers.

PROPOSITION 5.1. *(Constructor Lemma) The following are provable in* P2.

$$\vdash n \in \mathsf{N} \to \mathsf{s}\, n \in \mathsf{N} \quad \textit{and} \quad \vdash \mathsf{0} \in \mathsf{N}$$

Proof. Straightforward. The proof for successor appears as the top part of Figure 6. ∎

The inductive definitions, and the above proposition, do not depend on the particular value of the terms chosen for the constructors s and $\mathsf{0}$ and so on. They might even be uninterpreted constants. However, as we shall see, it turns out there is a good reason to choose them to be particular untyped lambda terms, namely the usual Church representations of successor and zero.

The deductive definition $\mathbf{N}$ is just the type of the Church numerals. It can be deduced from the inductive definition N by applying the Girard projection.

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

Furthermore, the constructors for the deductive definition can be deduced from the constructors for the inductive definition. If s and $\mathsf{0}$ are uninterpreted constants, then by the Constructor Lemma there are proofs in P2 of the following propositions; call these π_s and π_z.

$$\vdash n \in \mathsf{N} \to \mathsf{s}\, n \in \mathsf{N} \quad \text{and} \quad \vdash \mathsf{0} \in \mathsf{N}.$$

Applying the Girard projection to these proofs yields derivations in F2 of the following typings; call these $\delta_s = \pi_s^\circ$ and $\delta_z = \pi_z^\circ$.

$$\vdash \mathbf{s} : \mathbf{N} \to \mathbf{N} \quad \text{and} \quad \vdash \mathbf{0} : \mathbf{N}$$

Here $\mathbf{s}$ and $\mathbf{0}$ are the usual definitions for Church numerals.

$$\begin{array}{lcl} \mathbf{s} & = & \lambda n : \mathbf{N}.\, \Lambda X.\, \lambda s : X \to X.\, \lambda z : X.\, s\, (n\, X\, s\, z) \\ \mathbf{0} & = & \Lambda X.\, \lambda s : X \to X.\, \lambda z : X.\, z \end{array}$$

Figure 6 shows the proof π_s and the derivation δ_s in full.

In what follows, we assume $\mathsf{s} = |\mathbf{s}|$ and $\mathsf{0} = |\mathbf{0}|$. With this assumption, we will be able to show not only that the Girard projection takes the inductive definition into the deductive one, but also that, in the presence of parametricity, the Reynolds embedding takes the deductive definition into the inductive one.

Definition 5.2. We say that the natural numbers are *deductive* when $\mathsf{N} \subseteq \mathbf{N}^*$ and *inductive* when $\mathbf{N}^* \subseteq \mathsf{N}$.

We will show that the naturals are deductive and extensive, and that for the naturals parametricity and inductivity are equivalent.

Thus, in the presence of parametricity, the inductive and deductive definitions are equivalent. That is, one has not only $\mathbf{N} \equiv \mathsf{N}^\circ$ but also $\mathsf{N} = \mathbf{N}^*$, and hence not only $\mathbf{N} \equiv \mathbf{N}^{*\circ}$ but also $\mathsf{N} = \mathsf{N}^{\circ *}$. Thus there is a sense in which the Reynolds and Girard translations not only form an embedding-projection pair, but are truly inverses.

5.1. Naturals are extensive

As promised in Section 4, we show that the naturals are extensive, which corresponds, in a rough sense, to one half of Reynolds's Identity Extension Lemma for the naturals [Rey83]. We require extensionality.

PROPOSITION 5.3. *(Naturals are extensive) In* P2 + ext *we have*

$$\mathbf{N}^{\approx} \subseteq \mathbf{N}^{=}.$$

Proof.

$$\begin{array}{ll} & (n, n') \in \mathbf{N}^{\approx} \\ \equiv & \quad \text{(definition deductive naturals, parametric closure)} \\ & \forall \hat{X}.\, \forall s, s'.\, (\forall m, m'.\, (m, m') \in \hat{X} \to (s\, m, s'\, m') \in \hat{X}) \\ & \qquad \to \forall z, z'.\, (z, z') \in \hat{X} \to (n\, s\, z, n'\, s'\, z') \in \hat{X} \\ \Rightarrow & \quad \text{(instantiate } \hat{X} := \{\, (z, z') \mid z = z' \,\}) \\ & \forall s, s'.\, (\forall m, m'.\, m = m' \to s\, m = s'\, m') \to \forall z, z'.\, z = z' \to n\, s\, z = n'\, s'\, z' \\ \Rightarrow & \quad \text{(extensionality)} \\ & \forall s, s'.\, s = s' \to \forall z, z'.\, z = z' \to n\, s\, z = n'\, s'\, z' \\ \Rightarrow & \quad \text{(extensionality)} \\ & n = n' \end{array}$$

∎

5.2. Naturals are deductive

We show that the naturals are deductive. We do not require extensionality.

PROPOSITION 5.4. *(Naturals are deductive) In* P2 *we have*

$$\mathsf{N} \subseteq \mathbf{N}^{*}.$$

Proof.

$$\begin{array}{ll} & n \in \mathsf{N} \\ \equiv & \quad \text{(definition inductive naturals)} \\ & \forall X.\, (\forall m.\, m \in X \to \mathsf{s}\, m \in X) \to \mathsf{0} \in X \to n \in X \\ \Rightarrow & \quad \text{(instantiate } X := \mathbf{N}^{*}) \\ & (\forall m.\, m \in \mathbf{N}^{*} \to \mathsf{s}\, m \in \mathbf{N}^{*}) \to \mathsf{0} \in \mathbf{N}^{*} \to n \in \mathbf{N}^{*} \\ \Rightarrow & \quad \text{(Reynolds embedding applied to } \mathbf{s} \text{ and } \mathbf{0}) \\ & n \in \mathbf{N}^{*} \end{array}$$

∎

5.3. Inductive implies parametric

If the naturals are inductive then they are also parametric. Extensionality is not required.

PROPOSITION 5.5. *(Inductive implies parametric) In* P2 *we have*

$$\mathbf{N}^{*} \subseteq \mathsf{N} \quad \textit{implies} \quad \mathbf{N}^{=} \subseteq \mathbf{N}^{\approx}.$$

Proof.

$$\begin{array}{ll} & n \in \mathbf{N}^{*} \\ \Rightarrow & \quad \text{(assumption)} \\ & n \in \mathsf{N} \\ \equiv & \quad \text{(definition inductive naturals)} \\ & \forall X.\, (\forall m.\, m \in X \to \mathsf{s}\, m \in X) \to \mathsf{0} \in X \to n \in X \\ \Rightarrow & \quad \text{(instantiate } X := \{\, z \mid (z, z) \in \mathbf{N}^{\approx} \,\}) \\ & (\forall m.\, (m, m) \in \mathbf{N}^{\approx} \to (\mathsf{s}\, m, \mathsf{s}\, m) \in \mathbf{N}^{\approx}) \to (\mathsf{0}, \mathsf{0}) \in \mathbf{N}^{\approx} \to (n, n) \in \mathbf{N}^{\approx} \\ \Rightarrow & \quad \text{(Abstraction Theorem applied to } \mathbf{s} \text{ and } \mathbf{0}) \\ & (n, n) \in \mathbf{N}^{\approx} \end{array}$$

∎

Reynolds and Plotkin [RP90] were the first to suggest that parametricity implies inductivity, and Hasegawa [Has94] was the first to suggest the converse.

5.4. Parametric implies inductive

If the naturals are parametric then they are also inductive. Extensionality is required.

PROPOSITION 5.6. *(Parametric implies inductive) In* P2 + ext *we have*

$$\mathbf{N}^{=} \subseteq \mathbf{N}^{\approx} \quad \textit{implies} \quad \mathbf{N}^{*} \subseteq \mathsf{N}.$$

The proof depends on the following lemma. Böhm and Berarducci [BB85, Theorem 7.3] prove a similar result, though in a different framework and with a different technique.

PROPOSITION 5.7. *(Böhm and Berarducci's Lemma) In* P2 + ext *we have*

$$(n, n') \in \mathbf{N}^{\approx} \quad \textit{implies} \quad n\,\mathsf{s}\,\mathsf{0} = n'.$$

Proof.

$$\begin{array}{ll}
& (n, n') \in \mathbf{N}^{\approx} \\
\equiv & \quad \text{(definition deductive naturals, parametric closure)} \\
& \forall \hat{X}.\, \forall s, s'.\, (\forall m, m'.\, (m, m') \in \hat{X} \to (s\, m, s'\, m') \in \hat{X}) \\
& \qquad \to \forall z, z'.\, (z, z') \in \hat{X} \to (n\, s\, z, n'\, s'\, z') \in \hat{X} \\
\Rightarrow & \quad \text{(instantiate } \hat{X} := \{\, (n, n') \mid n\, s\, z = n' \,\},\ s := \mathsf{s},\ z := \mathsf{0},\ s' := s,\ z' := z) \\
& (\forall m, m'.\, m\, s\, z = m' \to \mathsf{s}\, m\, s\, z = s\, m') \to \mathsf{0}\, s\, z = z \to n\, \mathsf{s}\, \mathsf{0}\, s\, z = n'\, s\, z \\
\Rightarrow & \quad \text{(definition } \mathsf{s},\ \mathsf{0},\ \text{beta reduction)} \\
& (\forall m, m'.\, m\, s\, z = m' \to s\, (m\, s\, z) = s\, m') \to z = z \to n\, \mathsf{s}\, \mathsf{0}\, s\, z = n'\, s\, z \\
\Rightarrow & \quad \text{(simplify)} \\
& n\, \mathsf{s}\, \mathsf{0}\, s\, z = n'\, s\, z \\
\Rightarrow & \quad \text{(extensionality)} \\
& n\, \mathsf{s}\, \mathsf{0} = n'
\end{array}$$

∎

We can now prove Proposition 5.6.

Proof.

$$\begin{array}{ll}
& n \in \mathbf{N}^{*} \\
\equiv & \quad \text{(definition deductive naturals, Reynolds embedding)} \\
& \forall X.\, \forall s.\, (\forall m.\, m \in X \to s\, m \in X) \to \forall z.\, z \in X \to n\, s\, z \in X \\
\Rightarrow & \quad \text{(instantiate } X := \mathsf{N},\ s := \mathsf{s},\ z := \mathsf{0}) \\
& (\forall m.\, m \in \mathsf{N} \to \mathsf{s}\, m \in \mathsf{N}) \to \mathsf{0} \in \mathsf{N} \to n\, \mathsf{s}\, \mathsf{0} \in \mathsf{N} \\
\Rightarrow & \quad \text{(Constructor Lemma)} \\
& n\, \mathsf{s}\, \mathsf{0} \in \mathsf{N} \\
\Rightarrow & \quad \text{(parametricity of naturals, Böhm and Berarducci's Lemma)} \\
& n \in \mathsf{N}
\end{array}$$

∎

This completes the proof that parametricity and inductiveness are equivalent for the naturals. We have the following corollary.

PROPOSITION 5.8. *(Girard-Reynolds isomorphism) We have*

$$\mathbf{N} \equiv \mathbf{N}^{*\circ}$$

and furthermore in P2 + ext *we have*

$$\mathbf{N}^{=} = \mathbf{N}^{\approx} \quad \textit{iff} \quad \mathsf{N} = \mathsf{N}^{\circ *}.$$

6. REALIZABILITY AND GIRARD'S REPRESENTATION THEOREM

We have seen that from a derivation of $t : \mathbf{N}$ in F2 the Reynolds embedding yields a proof of $|t| \in \mathbf{N}^*$ in P2, and then the Girard projection yields a derivation of $t : \mathbf{N}$ in F2 again. Conversely, from a proof of $M \in \mathsf{N}$ in P2 the Girard projection yields a derivation of $t : \mathbf{N}$ in F2 for some term t, and then the Reynolds embedding yields a proof of $|t| \in \mathbf{N}^*$ in P2 again. Further, in the previous section we have seen that the naturals are parametric if and only if $\mathsf{N} = \mathbf{N}^*$. In this section, we show that we also have $M = |t|$, further strengthening the sense in which the Reynolds embedding and the Girard projection are inverses. From this result we will derive Girard's Representation Theorem [Gir72, GLT89, Lei90].

The key to the proof is a realizability interpretation, similar to those studied by Krivine and Parigot [KP90] and Takeuti [Tak98]. As we shall see, the realizability interpretation is related to both the Girard projection and the Reynolds embedding.

Recall that the Girard projection takes a proof π of a judgement in P2,

$$\phi_1, \ldots, \phi_n \vdash \psi,$$

into a derivation π° of a typing in F2,

$$x_1 : \phi_1^\circ, \ldots, x_n : \phi_n^\circ \vdash u : \psi^\circ.$$

The realizability interpretation takes the same proof π into a proof $\pi^\lhd$ of another judgement in P2,

$$x_1 \in \phi_1^\lhd, \ldots, x_n \in \phi_n^\lhd \vdash |u| \in \psi^\lhd,$$

where $x_1, \ldots, x_n$ and u are the same in the typing and the judgement. Each formula ϕ maps into a predicate $\phi^\lhd$. The realizability interpretation is defined in Figure 7. Note that a formula involving the predicate variable X on individuals maps into a formula involving the renaming $\hat{X}$ acting on pairs, as introduced in Section 4.

The existence of the realizability interpretation corresponds to Krivine and Parigot's *Conservation Lemma*, and the mapping from proofs to proofs shown in Figure 7 amounts to a diagramatic display of their proof of that lemma.

The realizability interpretation preserves substitution for terms and predicates, so that

$$(\phi[x := M])^\lhd \equiv \phi^\lhd[x := M] \quad \text{and} \quad (\phi[X := \{\, x \mid \psi \,\}])^\lhd \equiv \phi[X := \{\, (x, z) \mid z \in \phi^\lhd \,\}].$$

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 6.1. *(Realizability and the Reynolds embedding)*
For all types A, we have $z' \in (z \in A^*)^\lhd \equiv (z, z') \in A^{*\ddagger}$.

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.

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

∎

Combining the above with the results of the previous section, we see that $n' \in (n \in \mathbf{N}^*)^\lhd$ implies $n = n'$ if the naturals are parametric. Next we give a similar result which does not depend on parametricity. The proof is similar to that of Böhm and Berarducci's Lemma.

Proposition 6.2. *(Krivine and Parigot's Lemma) If* P2 + ext *we have*

$$n' \in (n \in \mathsf{N})^\lhd \quad \textit{implies} \quad n = n'.$$

Proof.

$$
\begin{array}{ll}
 & n' \in (n \in \mathsf{N})^\lhd \\
\equiv & \quad \text{(definition inductive naturals)} \\
 & n' \in (\forall X.\, (\forall m.\, m \in X \to \mathsf{s}\, m \in X) \to \mathsf{0} \in X \to n \in X)^\lhd \\
\equiv & \quad \text{(definition realizability interpretation)} \\
 & \forall \hat{X}.\, \forall s'.\, (\forall m.\, \forall m'.\, (m, m') \in \hat{X} \to (\mathsf{s}\, m, s'\, m') \in \hat{X}) \\
 & \qquad \to \forall z'.\, (\mathsf{0}, z') \in \hat{X} \to (n, n'\, s'\, z') \in \hat{X} \\
\Rightarrow & \quad \text{(instantiate } \hat{X} := \{\, (n, n') \mid n\, s\, z = n' \,\},\ s' := s,\ z' := z) \\
 & (\forall m.\, \forall m'.\, m\, s\, z = m' \to \mathsf{s}\, m\, s\, z = s\, m') \\
 & \qquad \to \mathsf{0}\, s\, z = z \to n\, s\, z = n'\, s\, z \\
\Rightarrow & \quad \text{(definition } \mathsf{s}, \mathsf{0}, \text{ beta reduction)} \\
 & (\forall m, m'.\, m\, s\, z = m' \to s\, (m\, s\, z) = s\, m') \\
 & \qquad\quad \to z = z \to n\, s\, z = n'\, s\, z \\
\Rightarrow & \quad \text{(simplify)} \\
 & n\, s\, z = n'\, s\, z \\
\Rightarrow & \quad \text{(extensionality)} \\
 & n = n'
\end{array}
$$

∎

As a corollary, we have that the Girard projection takes every untyped term provably in the inductive naturals in P2 into a typed term in the deductive naturals in F2 that represents it.

Proposition 6.3. *(Value representation) If* π *is a proof in* P2 *of the judgement*

$$\vdash M \in \mathsf{N}$$

then π° *is a derivation in* F2 *of the typing*

$$\vdash t : \mathbf{N}$$

and $M = |t|$ *is provable in* P2 + ext. *We call* t *the* representation *of* M.

Proof. The realizability interpretation yields a proof $\pi^\lhd$ of $|t| \in (M \in \mathsf{N})^\lhd$, from which Krivine and Parigot's Lemma deduces $M = |t|$. ∎

For example, let $2 = \mathsf{s}\,(\mathsf{s}\,\mathsf{0})$, and say we have a proof π in P2 of the judgement $\vdash 2 \in \mathsf{N}$. Applying the Girard projection to this proof yields a proof π° in F2 of the judgement $\vdash \mathbf{2} : \mathbf{N}$. Here the representation $\mathbf{2}$ of 2 is determined by the structure of the proof π. It is easy to find a proof π such that $\mathbf{2}$ is the second Church numeral. Can we be sure that this is true for *any* such proof? Yes! This is ensured by value representation, which guarantees $|\mathbf{2}| = 2$.

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

Proposition 6.4. *(Representation Theorem) If* π *is a proof in* P2 *of the judgement*

$$\vdash \forall x.\, x \in \mathsf{N} \to L\, x \in \mathsf{N}$$

then π° *is a derivation in* F2 *of the judgement*

$$\vdash s : \mathbf{N} \to \mathbf{N}$$

Further, if $M \in \mathsf{N}$ *has representation* $t : \mathbf{N}$ *then* $L\,M \in \mathsf{N}$ *has representation* $st : \mathbf{N}$.

Proof. The realizability interpretation yields a proof $\pi^\lhd$ of the judgment

$$\vdash |s| \in (\forall x.\, x \in \mathsf{N} \to L\, x \in \mathsf{N})^\lhd$$

We then reason as follows.

$$\begin{array}{ll} & |s| \in (\forall x.\, x \in \mathsf{N} \to L\, x \in \mathsf{N})^\lhd \\ \equiv & \quad \text{(Definition realizability interpretation)} \\ & \forall x, x'.\, x' \in (x \in \mathsf{N})^\lhd \to |s|\, x' \in (L\, x \in \mathsf{N})^\lhd \\ \Rightarrow & \quad \text{(instantiate } x := M,\ x' := |t|) \\ & |t| \in (M \in \mathsf{N})^\lhd \to |s\, t| \in (L\, M \in \mathsf{N})^\lhd \\ \Rightarrow & \quad (M \text{ has representation } t) \\ & |s\, t| \in (L\, M \in \mathsf{N})^\lhd \\ \Rightarrow & \quad \text{(Krivine and Parigot's Lemma)} \\ & L\, M = |s\, t| \end{array}$$

∎

For example, let plus be a term (perhaps an uninterpreted symbol) satisfying the following equations.

$$\begin{array}{lcl} \mathsf{plus}\,\mathsf{0}\,n & = & n \\ \mathsf{plus}\,(\mathsf{s}\,m)\,n & = & \mathsf{s}\,(\mathsf{plus}\,m\,n) \end{array}$$

Say there is a proof π in P2 of the judgement

$$\vdash m \in \mathsf{N} \to n \in \mathsf{N} \to \mathsf{plus}\,m\,n \in \mathsf{N}.$$

Then applying the Girard projection to this proof yields a derivation π° in F2 of the judgement

$$\vdash \mathbf{plus} : \mathbf{N} \to \mathbf{N} \to \mathbf{N}.$$

Here the representation **plus** of plus is determined by the structure of the proof π. One such proof yields $\mathbf{plus} \equiv \lambda m : \mathbf{N}.\, \lambda n : \mathbf{N}.\, m\, \mathbf{N}\, \mathbf{s}\, n$, which does indeed compute sums. Can we be sure that *any* proof yields a function that computes sums? Yes! This is ensured by the representation theorem, which guarantees that if $m \in \mathsf{N}$ and $n \in \mathsf{N}$ then $|\mathbf{plus}|\, m\, n = \mathsf{plus}\, m\, n$.

This is remarkable. We start with a proof about untyped terms. The Girard projection throws away the untyped term — throws away all the first-order parts of the proof — and constructs a typed term from the second-order parts of the proof. Yet it is guaranteed that the erasure of the typed term is equivalent to the original untyped term! It almost seems like magic, and, as with the best of magic tricks, knowing how it is done just makes it better.

ACKNOWLEDGMENTS

My thanks to Andrew Pitts, Jon Riecke, and referees of POPL 1999, LICS 2000, TACS 2001, and *Information and Computation*; with particular thanks to Andrew Pitts and some of the referees for spotting errors.

REFERENCES

[ACC93] M. Abadi, L. Cardelli, and P.-L. Curien, Formal Parametric Polymorphism, *Theoretical Computer Science* 121(1–2):9–58, December 1993. (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.

[Bar80] H. Barendregt, *The Lambda Calculus*, North-Holland, 1980.

[Bar91] H. Barendregt, Introduction to generalized types systems, *Journal of Functional Programming*, 1(2):125–154, April 1991.

[BFSS90] E. S. Bainbridge, P. J. Freyd, A. Scedrov, and P. J. Scott, Functorial polymorphism, in G. Huet, editor, *Logical Foundations of Functional Programming*, pp. 315–330, Addison-Wesley, 1990.

[BB85] C. Böhm and A. Berarducci, Automatic synthesis of typed Λ-programs on term algebras, *Theoretical Computer Science*, 39(2–3):135–154, August 1985.

[BC88] V. Breazu-Tannen and T. Coquand, Extensional models for polymorphism, *Theoretical Computer Science*, 59:85–114, 1988.

[CF58] H. B. Curry and R. Feys, *Combinatory Logic*, North Holland, 1958.

[CH88] T. Coquand and G. Huet, The calculus of constructions, *Information and Computation*, 76:95–120, 1988.

[Con86] R. Constable, *et al.*, *Implementing mathematics with the Nuprl proof development system*, Prentice-Hall, 1986.

[deB70] N. G. de Bruijn, The mathematical language of AUTOMATH, its usage and some of its extensions, *Proceedings of the Symposium on Automatic Demonstration*, LNCS 125, Springer-Verlag, 1970.

[DM82] L. Damas and R. Milner, Principal type schemes for functional programs, *9'th Annual Symposium on Principles of Programming Languages*, Albuquerque, N.M., ACM, January 1982.

[Fre79] Gottlob Frege. Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought (1879), in Jan van Heijenoort, editor, *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*, Harvard University Press, 1967.

[Gir72] J.-Y. Girard, *Interprétation functionelle et élimination des coupures dans l'arithmétique d'ordre supérieure*, Ph.D. thesis, Université Paris VII, 1972.

[GLT89] J.-Y. Girard, Y. Lafont, and P. Taylor, *Proofs and Types*, Cambridge University Press, 1989.

[Has94] R. Hasegawa, Categorical data types in parametric polymorphism, *Mathematical Structures in Computer Science*, 4:71-109, (1994).

[Hin69] R. Hindley, The principal type scheme of an object in combinatory logic, *Trans. Am. Math. Soc.*, 146:29–60, December 1969.

[How80] W. A. Howard, The formulae-as-types notion of construction, in J. P. Seldin and J. R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*, Academic Press, 1980. (The original version was circulated privately in 1969.)

[KP90] J.-L. Krivine and M. Parigot, Programming with proofs, *J. Inf. Process. Cybern. (EIK)*, 26(3): 149–167, 1990. (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.)

[Lei83] D. Leivant, Reasoning about functional programs and complexity classes associated with type disciplines, *24'th Symposium on Foundations of Computer Science*, Washington D.C., pp. 460–469, IEEE, 1983.

[Lei90] D. Leivant, Contracting proofs to programs, in P. Odifreddi, editor, *Logic and Computer Science*, Academic Press, 1990.

[Mai91] H. Mairson, Outline of a proof theory of parametricity, in J. Hughes, editor, *5'th International Conference on Functional Programming Languages and Computer Architecture*, Cambridge, Massachusetts, LNCS 523, Springer-Verlag, August 1991.

[Mar82] P. Martin-Löf, Constructive mathematics and computer programming, *6'th International Congress for Logic, Methodology, and Philosophy of Science*, pp. 153–175, North Holland, 1982.

[Mil78] R. Milner, A theory of type polymorphism in programming, *Journal of Computers and Systems Science*, 17:348–375, 1978.

[Mog86] E. Moggi, Communication to the Types electronic forum, 10 February 1986.

[MR91] Q. Ma and J. C. Reynolds, Types, abstraction, and parametric polymorphism, part 2, in S. Brookes *et al.* editors, *Mathematical Foundations of Programming Semantics*, LNCS, Springer Verlag, 1991.

[Pit87] A. M. Pitts, Polymorphism Is Set Theoretic, Constructively, in D. H. Pitt and A. Poigné and D. E. Rydeheard, editors, *Category Theory and Computer Science*, pp. 12–39, Edinburgh, 1987.

[Pit89] A. M. Pitts, Non-trivial power types can't be subtypes of polymorphic types, in *4th Annual Symposium on Logic in Computer Science*, pp. 6–13, IEEE Computer Society Press, Washington, 1989.

[Pit98] A. M. Pitts, Parametric polymorphism and operational equivalence, technical report 453, Cambridge University Computer Laboratory, 1998.

[PA93] G. Plotkin and M. Abadi, A logic for parametric polymorphism, in M. Bezem and J. F. Groote, editors, *Typed Lambda Calculi and Applications*, LNCS 664, Springer-Verlag, pp. 361–375, March 1993.

[PAC94] G. Plotkin, M. Abadi, and L. Cardelli, Subtyping and parametricity, *9'th Annual Symposium on Logic in Computer Science*, pp. 310–319, July 1994.

[Rey74] J. C. Reynolds, Towards a theory of type structure, in B. Robinet, editor, *Colloque sur la Programmation*, LNCS 19, Springer-Verlag, 1974.

[Rey83] J. C. Reynolds, Types, abstraction, and parametric polymorphism, in R. E. A. Mason, editor, *Information Processing 83*, pp. 513–523, North-Holland, Amsterdam, 1983.

[Rey84] J. C. Reynolds, Polymorphism is not set theoretic, in Kahn, MacQueen, and Plotkin, editors, *Semantics of Data Types*, Sophia-Antipolis, France, pp. 145–156, LNCS 173, Springer-Verlag, 1984.

[Rey90] J. C. Reynolds, Introduction to Part II: Polymorphic Lambda Calculus, in G. Huet, editor, *Logical Foundations of Functional Programming*, Addison-Wesley, 1990.

[RP90] J. C. Reynolds and G. D. Plotkin, On Functors Expressible in the polymorphic typed lambda calculus, in G. Huet, editor, *Logical Foundations of Functional Programming*, pp. 127–152, Addison-Wesley, 1990.

[Str67] C. Strachey, Fundamental concepts in programming languages, Lecture notes, International Summer School in Computer Programming, Copenhagen, August 1967. Reprinted in *Higher-Order and Symbolic Computation* 13(1/2):11-49, May 2000.

[Tak98] Izumi Takeuti, An axiomatic system of parametricity, *Fundamenta Informaticae*, IOS Press, 33:397-432, 1998.

[Wad89] P. Wadler, Theorems for free!, *4'th International Conference on Functional Programming Languages and Computer Architecture*, ACM Press, London, September 1989.

[Wad91] P. Wadler, Recursive types for free!, manuscript, 1991.

[Wad01] P. Wadler, The Girard-Reynolds isomorphism, in Kobayashi and Pierce, editors, *Theoretical Aspects of Computer Software*, LNCS 2215, Springer Verlag, pp. 468–491, Sendai, Japan, October 2001.

Syntax

Type variables	X, Y, Z		
Individual variables	x, y, z		
Types	A, B	$::=$	$X \mid A \to B \mid \forall X.\, B$
Typed terms	s, t, u	$::=$	$x \mid \lambda x{:}A.\, u \mid s\ t \mid \Lambda X.\, u \mid s\ A$
Contexts	Γ	$::=$	$x_1 : A_1, \ldots, x_n : A_n$

Rules

$$\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i}\ \text{Id}$$

$$\frac{\Gamma,\, x : A \vdash u : B}{\Gamma \vdash \lambda x{:}A.\, u : A \to B}\ {\to}\text{-I} \qquad \frac{\Gamma \vdash s : A \to B \quad \Gamma \vdash t : A}{\Gamma \vdash s\ t : B}\ {\to}\text{-E}$$

$$\frac{\Gamma \vdash u : B}{\Gamma \vdash \Lambda X.\, u : \forall X.\, B}\ \forall^2\text{-I} \quad \text{when } X \text{ not free in } \Gamma \qquad \frac{\Gamma \vdash s : \forall X.\, B}{\Gamma \vdash s\ A : B[X := A]}\ \forall^2\text{-E}$$

Reductions

$$\frac{\dfrac{\Gamma,\, x : A \overset{\vdots\ \delta}{\vdash} u : B}{\Gamma \vdash \lambda x{:}A.\, u : A \to B}\ {\to}\text{-I} \qquad \Gamma \overset{\vdots\ \epsilon}{\vdash} t : A}{\Gamma \vdash (\lambda x{:}A.\, u)\ t : B}\ {\to}\text{-E} \quad \Rightarrow \quad \Gamma \overset{\vdots\ \delta[x := \epsilon]}{\vdash} u[x := t] : B$$

$$\frac{\dfrac{\Gamma \overset{\vdots\ \delta}{\vdash} u : B}{\Gamma \vdash \Lambda X.\, u : \forall X.\, B}\ \forall^2\text{-I}}{\Gamma \vdash (\Lambda X.\, u)\ A : B[X := A]}\ \forall^2\text{-E} \quad \Rightarrow \quad \Gamma \overset{\vdots\ \delta[X := A]}{\vdash} u[X := A] : B[X := A]$$

FIG. 1: Second-order lambda calculus (F2)

Syntax

Predicate variables	X, Y, Z		
Individual variables	x, y, z		
Terms	L, M, N	$::=$	$x \mid \lambda x.\, M \mid M\ N$
Formulas	ϕ, ψ	$::=$	$M \in X \mid \phi \to \psi \mid \forall x.\, \psi \mid \forall X.\, \psi$
Contexts	Θ	$::=$	$\phi_1, \ldots, \phi_n$

Rules

$$\frac{}{\phi_1, \ldots, \phi_n \vdash \phi_i}\ \mathrm{Id}$$

$$\frac{\Theta, \phi \vdash \psi}{\Theta \vdash \phi \to \psi}\ {\to}\text{-I} \qquad \frac{\Theta \vdash \phi \to \psi \quad \Theta \vdash \phi}{\Theta \vdash \psi}\ {\to}\text{-E}$$

$$\frac{\Theta \vdash \psi}{\Theta \vdash \forall x.\, \psi}\ \forall^1\text{-I} \quad \text{when } x \text{ not free in } \Theta \qquad \frac{\Theta \vdash \forall x.\, \psi}{\Theta \vdash \psi[x := M]}\ \forall^1\text{-E}$$

$$\frac{\Theta \vdash \psi}{\Theta \vdash \forall X.\, \psi}\ \forall^2\text{-I} \quad \text{when } X \text{ not free in } \Theta \qquad \frac{\Theta \vdash \forall X.\, \psi}{\Theta \vdash \psi[X := \{\, x \mid \phi \,\}]}\ \forall^2\text{-E}$$

$$\frac{\Theta \vdash \psi[x := M]}{\Theta \vdash \psi[x := N]}\ \beta \quad \text{when } M =_\beta N$$

Reductions

$$\frac{\dfrac{\overset{\vdots\ \pi}{\Theta, \phi \vdash \psi}}{\Theta \vdash \phi \to \psi}\ {\to}\text{-I} \qquad \overset{\vdots\ \rho}{\Theta \vdash \phi}}{\Theta \vdash \psi}\ {\to}\text{-E} \quad \Rightarrow \quad \overset{\vdots\ \pi[\phi := \rho]}{\Theta \vdash \psi}$$

$$\frac{\dfrac{\overset{\vdots\ \pi}{\Theta \vdash \psi}}{\Theta \vdash \forall x.\, \psi}\ \forall\text{-I}}{\Theta \vdash \psi[x := M]}\ \forall\text{-E} \quad \Rightarrow \quad \overset{\vdots\ \pi[x := M]}{\Theta \vdash \psi[x := M]}$$

$$\frac{\dfrac{\overset{\vdots\ \pi}{\Theta \vdash \psi}}{\Theta \vdash \forall X.\, \psi}\ \forall^2\text{-I}}{\Theta \vdash \psi[X := \{\, x \mid \phi \,\}]}\ \forall^2\text{-E} \quad \Rightarrow \quad \overset{\vdots\ \pi[X := \{\, x \mid \phi \,\}]}{\Theta \vdash \psi[X := \{\, x \mid \phi \,\}]}$$

FIG. 2: Second-order propositional logic (P2)

Terms

$$\begin{array}{lcl} |x| & \equiv & x \\ |\lambda x{:}A.\, u| & \equiv & \lambda x.\, |u| \\ |s\, t| & \equiv & |s|\, |t| \\ |\Lambda X.\, u| & \equiv & |u| \\ |s\, A| & \equiv & |s| \end{array}$$

Types

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

Contexts

$$(x_1 : A_1,\, \ldots,\, x_n : A_n)^* \quad \equiv \quad x_1 \in A_1^*,\, \ldots,\, x_n \in A_n^*$$

Judgements

$$(\Gamma \vdash t : A)^* \quad \equiv \quad \Gamma^* \vdash |t| \in A^*$$

Proofs

$$\left(\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i}\,\mathrm{Id}\right)^* \quad \equiv \quad \frac{}{x_1 \in A_1^*, \ldots, x_n \in A_n^* \vdash x_i \in A_i^*}\,\mathrm{Id}$$

$$\left(\frac{\Gamma,\, x : A \overset{\vdots\,\delta}{\vdash} u : B}{\Gamma \vdash \lambda x : A.\, u : A \to B}\,{\to}\text{-I}\right)^* \quad \equiv \quad \frac{\dfrac{\dfrac{\Gamma^*,\, x \in A^* \overset{\vdots\,\delta^*}{\vdash} |u| \in B^*}{\Gamma^*,\, x \in A^* \vdash (\lambda x.\, |u|)\, x \in B^*}\,\beta}{\Gamma^* \vdash x \in A^* \to (\lambda x.\, |u|)\, x \in B^*}\,{\to}\text{-I}}{\Gamma^* \vdash \forall x.\, x \in A^* \to (\lambda x.\, |u|)\, x \in B^*}\,\forall^1\text{-I}$$

$$\left(\frac{\Gamma \vdash s : A \to B \overset{\vdots\,\delta}{} \qquad \Gamma \vdash t : A \overset{\vdots\,\epsilon}{}}{\Gamma \vdash s\, t : B}\,{\to}\text{-E}\right)^*$$

$$\equiv \quad \frac{\dfrac{\Gamma^* \vdash \forall x.\, x \in A^* \to |s|\, x \in B^* \overset{\vdots\,\delta^*}{}}{\Gamma^* \vdash |t| \in A^* \to |s|\, |t| \in B^*}\,\forall^1\text{-E} \qquad \Gamma^* \vdash |t| \overset{\vdots\,\epsilon^*}{\in} A^*}{\Gamma^* \vdash |s|\, |t| \in B^*}\,{\to}\text{-E}$$

$$\left(\frac{\Gamma \vdash u : B \overset{\vdots\,\delta}{}}{\Gamma \vdash \Lambda X.\, u : \forall X.\, B}\,\forall^2\text{-I}\right)^* \quad \equiv \quad \frac{\Gamma^* \vdash |u| \overset{\vdots\,\delta^*}{\in} B^*}{\Gamma^* \vdash \forall X.\, |u| \in B^*}\,\forall^2\text{-I}$$

$$\left(\frac{\Gamma \vdash s : \forall X.\, B \overset{\vdots\,\delta}{}}{\Gamma \vdash s\, A : B[X := A]}\,\forall^2\text{-E}\right)^* \quad \equiv \quad \frac{\Gamma^* \vdash \forall X.\, |s| \overset{\vdots\,\delta^*}{\in} B^*}{\Gamma^* \vdash |s| \in B^*[X := A^*]}\,\forall^2\text{-E}$$

FIG. 3: The Reynolds embedding

Formulas

$$\begin{array}{lcl}(M \in X)^\circ & \equiv & X \\ (\phi \to \psi)^\circ & \equiv & \phi^\circ \to \psi^\circ \\ (\forall x.\, \psi)^\circ & \equiv & \psi^\circ \\ (\forall X.\, \psi)^\circ & \equiv & \forall X.\, \psi^\circ\end{array}$$

Contexts

$$(\phi_1, \ldots, \phi_n)^\circ \quad \equiv \quad x_1 : \phi_1^\circ, \ldots, x_n : \phi_n^\circ$$

Judgements

$$(\Theta \vdash \phi)^\circ \quad \equiv \quad \Theta^\circ \vdash t : \phi^\circ$$

Proofs

$$\left(\frac{}{\phi_1, \ldots, \phi_n \vdash \phi_i}\,\mathrm{Id}\right)^\circ \quad \equiv \quad \frac{}{x_1 : \phi_1^\circ, \ldots, x_n : \phi_n^\circ \vdash x_i : \phi_i^\circ}\,\mathrm{Id}$$

$$\left(\frac{\begin{array}{c}\vdots\,\pi \\ \Theta,\, \phi \vdash \psi\end{array}}{\Theta \vdash \phi \to \psi}\,{\to}\text{-I}\right)^\circ \quad \equiv \quad \frac{\begin{array}{c}\vdots\,\pi^\circ \\ \Theta^\circ,\, x : \phi^\circ \vdash u : \psi^\circ\end{array}}{\Theta^\circ \vdash \lambda x{:}\phi^\circ.\, u : \phi^\circ \to \psi^\circ}\,{\to}\text{-I}$$

$$\left(\frac{\begin{array}{cc}\vdots\,\pi & \vdots\,\rho \\ \Theta \vdash \phi \to \psi & \Theta \vdash \phi\end{array}}{\Theta \vdash \psi}\,{\to}\text{-E}\right)^\circ \quad \equiv \quad \frac{\begin{array}{cc}\vdots\,\pi^\circ & \vdots\,\rho^\circ \\ \Theta^\circ \vdash s : \phi^\circ \to \psi^\circ & \Theta^\circ \vdash t : \phi^\circ\end{array}}{\Theta^\circ \vdash s\, t : \psi^\circ}\,{\to}\text{-E}$$

$$\left(\frac{\begin{array}{c}\vdots\,\pi \\ \Theta \vdash \psi\end{array}}{\Theta \vdash \forall x.\, \psi}\,\forall^1\text{-I}\right)^\circ \quad \equiv \quad \begin{array}{c}\vdots\,\pi^\circ \\ \Theta^\circ \vdash u : \psi^\circ\end{array}$$

$$\left(\frac{\begin{array}{c}\vdots\,\pi \\ \Theta \vdash \forall x.\, \psi\end{array}}{\Theta \vdash \psi[x := M]}\,\forall^1\text{-E}\right)^\circ \quad \equiv \quad \begin{array}{c}\vdots\,\pi^\circ \\ \Theta^\circ \vdash s : \psi^\circ\end{array}$$

$$\left(\frac{\begin{array}{c}\vdots\,\pi \\ \Theta \vdash \psi\end{array}}{\Theta \vdash \forall X.\, \psi}\,\forall^2\text{-I}\right)^\circ \quad \equiv \quad \frac{\begin{array}{c}\vdots\,\pi^\circ \\ \Theta^\circ \vdash u : \psi^\circ\end{array}}{\Theta^\circ \vdash \Lambda X.\, u : \forall X.\, \psi^\circ}\,\forall^2\text{-I}$$

$$\left(\frac{\begin{array}{c}\vdots\,\pi \\ \Theta \vdash \forall X.\, \psi\end{array}}{\Theta \vdash \psi[X := \{\, x \mid \phi \,\}]}\,\forall^2\text{-E}\right)^\circ \quad \equiv \quad \frac{\begin{array}{c}\vdots\,\pi^\circ \\ \Theta^\circ \vdash s : \forall X.\, \psi^\circ\end{array}}{\Theta^\circ \vdash s\, \phi^\circ : \psi^\circ[X := \phi^\circ]}\,\forall^2\text{-E}$$

$$\left(\frac{\begin{array}{c}\vdots\,\pi \\ \Theta \vdash \psi[x := M]\end{array}}{\Theta \vdash \psi[x := N]}\,\beta\right)^\circ \quad \equiv \quad \begin{array}{c}\vdots\,\pi^\circ \\ \Theta^\circ \vdash \psi^\circ\end{array}$$

FIG. 4: The Girard projection

Formulas

$$\begin{array}{lcl}(M \in X)^\ddagger & \equiv & (M, M') \in \hat{X} \\ (\phi \to \psi)^\ddagger & \equiv & \phi^\ddagger \to \psi^\ddagger \\ (\forall x.\, \psi)^\ddagger & \equiv & \forall x.\, \forall x'.\, \psi^\ddagger \\ (\forall X.\, \psi)^\ddagger & \equiv & \forall \hat{X}.\, \psi^\ddagger \end{array}$$

Contexts

$$(\phi_1, \ldots, \phi_n)^\ddagger \quad \equiv \quad \phi_1^\ddagger, \ldots, \phi_n^\ddagger$$

Judgements

$$(\Theta \vdash \phi)^\ddagger \quad \equiv \quad \Theta^\ddagger \vdash \phi^\ddagger$$

Proofs

$$\left(\frac{}{\phi_1, \ldots, \phi_n \vdash \phi_i}\,\mathrm{Id}\right)^\ddagger \quad \equiv \quad \frac{}{\phi_1^\ddagger, \ldots, \phi_n^\ddagger \vdash \phi_i^\ddagger}\,\mathrm{Id}$$

$$\left(\frac{\begin{array}{c}\vdots\,\pi \\ \Theta,\, \phi \vdash \psi\end{array}}{\Theta \vdash \phi \to \psi}\,{\to}\text{-I}\right)^\ddagger \quad \equiv \quad \frac{\begin{array}{c}\vdots\,\pi^\ddagger \\ \Theta^\ddagger,\, \phi^\ddagger \vdash \psi^\ddagger\end{array}}{\Theta^\ddagger \vdash \phi^\ddagger \to \psi^\ddagger}\,{\to}\text{-I}$$

$$\left(\frac{\begin{array}{c}\vdots\,\pi \\ \Theta \vdash \phi \to \psi\end{array} \qquad \begin{array}{c}\vdots\,\rho \\ \Theta \vdash \phi\end{array}}{\Theta \vdash \psi}\,{\to}\text{-E}\right)^\ddagger \quad \equiv \quad \frac{\begin{array}{c}\vdots\,\pi^\ddagger \\ \Theta^\ddagger \vdash \phi^\ddagger \to \psi^\ddagger\end{array} \qquad \begin{array}{c}\vdots\,\rho^\ddagger \\ \Theta^\ddagger \vdash \phi^\ddagger\end{array}}{\Theta^\ddagger \vdash \psi^\ddagger}\,{\to}\text{-E}$$

$$\left(\frac{\begin{array}{c}\vdots\,\pi \\ \Theta \vdash \psi\end{array}}{\Theta \vdash \forall x.\, \psi}\,\forall^1\text{-I}\right)^\ddagger \quad \equiv \quad \frac{\begin{array}{c}\vdots\,\pi^\ddagger \\ \Theta^\ddagger \vdash \psi^\ddagger\end{array}}{\Theta^\ddagger \vdash \forall x.\, \forall x'.\, \psi^\ddagger}\,\forall^1\text{-I (twice)}$$

$$\left(\frac{\begin{array}{c}\vdots\,\pi \\ \Theta \vdash \forall x.\, \psi\end{array}}{\Theta \vdash \psi[x := M]}\,\forall^1\text{-E}\right)^\ddagger \quad \equiv \quad \frac{\begin{array}{c}\vdots\,\pi^\ddagger \\ \Theta^\ddagger \vdash \forall x.\, \forall x'.\, \psi^\ddagger\end{array}}{\Theta^\ddagger \vdash \psi^\ddagger[x := M, x' := M']}\,\forall^1\text{-E (twice)}$$

$$\left(\frac{\begin{array}{c}\vdots\,\pi \\ \Theta \vdash \psi\end{array}}{\Theta \vdash \forall X.\, \psi}\,\forall^2\text{-I}\right)^\ddagger \quad \equiv \quad \frac{\begin{array}{c}\vdots\,\pi^\ddagger \\ \Theta^\ddagger \vdash \psi^\ddagger\end{array}}{\Theta^\ddagger \vdash \forall \hat{X}.\, \psi^\ddagger}\,\forall^2\text{-I}$$

$$\left(\frac{\begin{array}{c}\vdots\,\pi \\ \Theta \vdash \forall X.\, \psi\end{array}}{\Theta \vdash \psi[X := \{\, x \mid \phi \,\}]}\,\forall^2\text{-E}\right)^\ddagger \quad \equiv \quad \frac{\begin{array}{c}\vdots\,\pi^\ddagger \\ \Theta^\ddagger \vdash \forall \hat{X}.\, \psi^\ddagger\end{array}}{\Theta^\ddagger \vdash \psi^\ddagger[\hat{X} := \{\, (x, x') \mid \phi^\ddagger \,\}]}\,\forall^2\text{-E}$$

$$\left(\frac{\begin{array}{c}\vdots\,\pi \\ \Theta \vdash \psi[x := M]\end{array}}{\Theta \vdash \psi[x := N]}\,\beta\right)^\ddagger \quad \equiv \quad \frac{\begin{array}{c}\vdots\,\pi^\ddagger \\ \Theta^\ddagger \vdash \psi^\ddagger[x := M,\, x' := M']\end{array}}{\Theta^\ddagger \vdash \psi^\ddagger[x := N,\, x' := N']}\,\beta\text{ (twice)}$$

FIG. 5: The doubling embedding

$$
\left(
\dfrac{
\dfrac{\dfrac{}{\Theta \vdash \phi_s}\,\mathrm{Id}}{\dfrac{}{\Theta \vdash n \in X \to \mathsf{s}\, n \in X}\,\forall^1\text{-E}}
\qquad
\dfrac{\dfrac{}{\Theta \vdash n \in \mathsf{N}}\,\mathrm{Id}}{\dfrac{}{\Theta \vdash \phi_s \to \phi_z \to n \in X}\,\forall^2\text{-E}}
\quad
\dfrac{}{\Theta \vdash \phi_s}\,\mathrm{Id}
\quad
\dfrac{}{\Theta \vdash \phi_z}\,\mathrm{Id}
}{\dfrac{}{\Theta \vdash n \in X}\,\to\text{-E (twice)}}\,\to\text{-E}
}{\dfrac{\dfrac{\dfrac{n \in \mathsf{N}, \phi_s, \phi_z \vdash \mathsf{s}\, n \in X}{n \in \mathsf{N} \vdash \phi_s \to \phi_z \to \mathsf{s}\, n \in X}\,\to\text{-I (twice)}}{n \in \mathsf{N} \vdash \mathsf{s}\, n \in \mathsf{N}}\,\forall^2\text{-I}}{\vdash n \in \mathsf{N} \to \mathsf{s}\, n \in \mathsf{N}}\,\to\text{-I}}
\right)^{\circ}
$$

$$
\equiv
\dfrac{
\dfrac{\dfrac{}{\Gamma \vdash s : X \to X}\,\mathrm{Id}
\qquad
\dfrac{\dfrac{\dfrac{}{\Gamma \vdash n : \mathbf{N}}\,\mathrm{Id}}{\dfrac{}{\Gamma \vdash n\, X : A_s \to A_z \to X}\,\forall^2\text{-E}}
\quad
\dfrac{}{\Gamma \vdash s : A_s}\,\mathrm{Id}
\quad
\dfrac{}{\Gamma \vdash z : A_z}\,\mathrm{Id}}{\dfrac{}{\Gamma \vdash n\, X\, s\, z : X}\,\to\text{-E (twice)}}\,\to\text{-E}}{\dfrac{\dfrac{\dfrac{n : \mathbf{N},\, s : A_s,\, z : A_z \vdash s\,(n\, X\, s\, z) : X}{n : \mathbf{N} \vdash \lambda s : A_s.\, \lambda z : A_z.\, s\,(n\, X\, s\, z) : A_s \to A_z \to X}\,\to\text{-I (twice)}}{n : \mathbf{N} \vdash \Lambda X.\, \lambda s : A_s.\, \lambda z : A_z.\, s\,(n\, X\, s\, z) : \mathbf{N}}\,\forall^2\text{-I}}{\vdash \lambda n : \mathbf{N}.\, \Lambda X.\, \lambda s : X \to X.\, \lambda z : X.\, s\,(n\, X\, s\, z) : \mathbf{N} \to \mathbf{N}}\,\to\text{-I}
$$

$$
\begin{array}{lcllcl}
\phi_s & \equiv & \forall m.\, m \in X \to \mathsf{s}\, m \in X & A_s & \equiv & X \to X \\
\phi_z & \equiv & \mathsf{0} \in X & A_z & \equiv & X \\
\Theta & \equiv & n \in \mathsf{N},\, \phi_s,\, \phi_z & \Gamma & \equiv & n : \mathbf{N},\, s : A_s,\, z : A_z
\end{array}
$$

FIG. 6: Successor

Formulas

$$\begin{array}{lcl}(M \in X)^{\lhd} & \equiv & \{\, z \,|\, (M, z) \in \hat{X} \,\} \\ (\phi \to \psi)^{\lhd} & \equiv & \{\, z \,|\, \forall x'.\, x' \in \phi^{\lhd} \to z\, x' \in \psi^{\lhd} \,\} \\ (\forall x.\, \psi)^{\lhd} & \equiv & \{\, z \,|\, \forall x.\, z \in \psi^{\lhd} \,\} \\ (\forall X.\, \psi)^{\lhd} & \equiv & \{\, z \,|\, \forall \hat{X}.\, z \in \psi^{\lhd} \,\}\end{array}$$

Contexts

$$(\phi_1, \ldots, \phi_n)^{\lhd} \quad \equiv \quad x_1 \in \phi_1^{\lhd}, \ldots, x_n \in \phi_n^{\lhd}$$

Judgements

$$(\Theta \vdash \phi)^{\lhd} \quad \equiv \quad \Theta^{\lhd} \vdash |t| \in \phi^{\lhd}$$

Proofs

$$\left(\frac{}{\phi_1, \ldots, \phi_n \vdash \phi_i}\,\mathrm{Id}\right)^{\lhd} \quad \equiv \quad \frac{}{x_1 \in \phi_1^{\lhd}, \ldots, x_n \in \phi_n^{\lhd} \vdash x_i \in \phi_i^{\lhd}}\,\mathrm{Id}$$

$$\left(\frac{\Theta,\, \phi \overset{\vdots\ \pi}{\vdash} \psi}{\Theta \vdash \phi \to \psi}\,{\to}\text{-I}\right)^{\lhd} \quad \equiv \quad \frac{\dfrac{\Theta^{\lhd},\, x \in \phi^{\lhd} \overset{\vdots\ \pi^{\lhd}}{\vdash} |u| \in \psi^{\lhd}}{\Theta^{\lhd},\, x \in \phi^{\lhd} \vdash (\lambda x.\, |u|)\, x \in \psi^{\lhd}}\,\beta}{\Theta^{\lhd} \vdash x \in \phi^{\lhd} \to (\lambda x.\, |u|)\, x \in \psi^{\lhd}}\,{\to}\text{-I}}{\Theta^{\lhd} \vdash \forall x.\, x \in \phi^{\lhd} \to (\lambda x.\, |u|)\, x \in \psi^{\lhd}}\,\forall^1\text{-I}$$

$$\left(\frac{\Theta \overset{\vdots\ \pi}{\vdash} \phi \to \psi \qquad \Theta \overset{\vdots\ \rho}{\vdash} \phi}{\Theta \vdash \psi}\,{\to}\text{-E}\right)^{\lhd}$$

$$\equiv \quad \frac{\dfrac{\Theta^{\lhd} \overset{\vdots\ \pi^{\lhd}}{\vdash} \forall x.\, x \in \phi^{\lhd} \to |s|\, x \in \psi^{\lhd}}{\Theta^{\lhd} \vdash |t| \in \phi^{\lhd} \to |s|\, |t| \in \psi^{\lhd}}\,\forall^1\text{-E} \qquad \Theta^{\lhd} \overset{\vdots\ \rho^{\lhd}}{\vdash} |t| \in \phi^{\lhd}}{\Theta^{\lhd} \vdash |s|\, |t| : \psi^{\circ}}\,{\to}\text{-E}$$

$$\left(\frac{\Theta \overset{\vdots\ \pi}{\vdash} \psi}{\Theta \vdash \forall x.\, \psi}\,\forall^1\text{-I}\right)^{\lhd} \quad \equiv \quad \frac{\Theta^{\lhd} \overset{\vdots\ \pi^{\lhd}}{\vdash} |u| \in \psi^{\lhd}}{\Theta^{\lhd} \vdash \forall x.\, |u| \in \psi^{\lhd}}\,\forall^1\text{-I}$$

$$\left(\frac{\Theta \overset{\vdots\ \pi}{\vdash} \forall x.\, \psi}{\Theta \vdash \psi[x := M]}\,\forall^1\text{-E}\right)^{\lhd} \quad \equiv \quad \frac{\Theta^{\lhd} \overset{\vdots\ \pi^{\lhd}}{\vdash} \forall x.\, |u| \in \psi^{\lhd}}{\Theta^{\lhd} \vdash |u| \in \psi^{\lhd}[x := M]}\,\forall^1\text{-E}$$

$$\left(\frac{\Theta \overset{\vdots\ \pi}{\vdash} \psi}{\Theta \vdash \forall X.\, \psi}\,\forall^2\text{-I}\right)^{\lhd} \quad \equiv \quad \frac{\Theta^{\lhd} \overset{\vdots\ \pi^{\lhd}}{\vdash} |u| \in \psi^{\lhd}}{\Theta^{\lhd} \vdash \forall \hat{X}.\, |u| \in \psi^{\lhd}}\,\forall^2\text{-I}$$

$$\left(\frac{\Theta \overset{\vdots\ \pi}{\vdash} \forall X.\, \psi}{\Theta \vdash \psi[X := \{\, x \,|\, \phi \,\}]}\,\forall^2\text{-E}\right)^{\lhd} \quad \equiv \quad \frac{\Theta^{\lhd} \overset{\vdots\ \pi^{\lhd}}{\vdash} \forall \hat{X}.\, |s| \in \psi^{\lhd}}{\Theta^{\lhd} \vdash |s| \in \psi^{\lhd}[\hat{X} := \{\, (x, z) \,|\, z \in \phi^{\lhd} \,\}]}\,\forall^2\text{-E}$$

$$\left(\frac{\Theta \overset{\vdots\ \pi}{\vdash} \psi[x := M]}{\Theta \vdash \psi[x := N]}\,\beta\right)^{\lhd} \quad \equiv \quad \frac{\Theta^{\lhd} \overset{\vdots\ \pi^{\lhd}}{\vdash} |t| \in \psi^{\lhd}[x := M]}{\Theta^{\lhd} \vdash |t| \in \psi^{\lhd}[x := N]}\,\beta$$

FIG. 7: The realizability interpretation