Fundamenta Informaticae XX (1998) 1–29
IOS Press

An Axiomatic System of Parametricity

Izumi Takeuti
Department of Information Science
Kyoto University
Kyoto, 606-8501 Japan
takeuti@kuis.kyoto-u.ac.jp

Abstract. Plotkin and Abadi have proposed a syntactic system for parametricity based on a second order predicate logic. This paper shows three theorems about that system. The first is consistency of the system, which is proved by the method of relativization. The second is that polyadic parametricities of recursive types are equivalent to each other. The third is that the theory of parametricity for recursive types is self-realizable. As a corollary of the third theorem, the theory of parametricity for recursive types satisfies the term extraction property.

Keywords: parametricity, polymorphism, typed lambda calculus

1. Introduction

1.1. Related Works

There are many works on parametric polymorphism. Interest in parametricity arose in two areas, model theory and formal logic. The original meaning of parametricity was as a notion for models of polymorphic calculi, such as System F, the second order lambda calculus studied by Girard [4]. More recently, many researchers have shown interest in parametricity in formal logic. This paper discusses parametricity in formal logic.

In the sense of Reynolds [9], parametricity is the property that if X is of universal type $\Pi\gamma.F[\gamma]$ then $XA(F[R])XB$ holds for all types A and B and for all relations R between A and B. In the original meaning, parametricity was a notion for universal types. This paper regards the parametricity notion as extended into the notion for all types, just as Plotkin and Abadi [8] do. The parametricity notion for functional types is called naturality by Hasegawa [6, 7].

Abadi, Cardelli and Curien [1] propose System R, a formal system for parametricity. It is a logical system for binary relations between terms of System F. Because it deals only with binary predicates, its expressive power is rather weak.

Hasegawa [6, 7] makes a category-like model for polymorphism, which is called a relation-frame. A relation-frame is a model of System F, and if it is parametric, then it is a model for System R. The informal logic which is used by Hasegawa [6, 7] is second order predicate logic, and formalizing his informal logic, we can obtain a formal treatment of parametricity in formal second order predicate logic. Hasegawa [6, 7] does not show the existence of the relation-frame which is parametric for all the types of System F, although he shows the existence of the relation-frame which is parametric for some particular types of System F, such as pair types, sum types, natural numbers, and so forth.

Bellucci, Abadi, and Curien [3] construct a model of System R from a partial equivalence relation model of System F. The existence of the model gives a model theoretic proof of the consistency of the theory of parametricity.

Plotkin and Abadi [8] formalize parametricity based on a second order predicate logic. In their system, the basic logic is not specialized to the treatment of parametricity, and the axioms implement the parametricity. They also discuss polyadic parametricities. The original notion of parametricity was a notion for binary predicates. Ternary parametricity for a universal type is the property that if X is of universal type $\Pi\gamma.F[\gamma]$ then it holds that $F[R](XA)(XB)(XC)$ for all types A, B and C and for all predicates R for terms of type A, B and C. Quaternary parametricity and n-ary parametricity for general n are defined in a similar way. Plotkin and Abadi propose a problem: Are the theory of binary parametricity and the theory of ternary parametricity equivalent? How about for quaternary, or n-ary for general n? We solve a part of this problem in this paper, which is an extension of their work.

1.2. Outline

We shall give three main results in this paper. The first will be the consistency of the theory for parametricity. This theory will be given by an axiomatic system of parametricity on a basic logic **L**. The basic logic **L** is second order predicate logic over terms of System F. The consistency has been already shown by the model theoretic proof in [3], but there has been no syntactic proof of consistency to the best of the author's knowledge. Our proof will be a direct proof, which uses relativization. It will be given in Section 3.

The second result will be that the theories of polyadic parametricities are equivalent to each other as long as types are restricted to recursive types. Recursive types will be defined as

$$\mathsf{R} ::= \mathbf{1} \mid \gamma \mid \mathsf{R} \to \mathsf{R} \mid \mathsf{R} \times \mathsf{R} \mid \mathsf{R} + \mathsf{R} \mid \mu\gamma.\mathsf{R} \mid \nu\gamma.\mathsf{R}.$$

In this definition, the constructors are encoded as the types of the system F. In $\mu\gamma.T$ and $\nu\gamma.T$, the type variable γ occurs only positively. We shall define an axiom $\mathbf{Par}_n(T)$ which asserts the n-ary parametricity of a type T, and an axiomatic system $\mathbf{Par}_n(\mathsf{R})$ as the set of axioms $\mathbf{Par}_n(T)$ for all the types T in R. We shall prove that the theories of $\mathbf{Par}_m(\mathsf{R})$ and of $\mathbf{Par}_n(\mathsf{R})$ are equal to each other for arbitrary numbers $m, n \geq 2$. This will be a partial answer to the problem of Plotkin and Abadi [8], and will be given in Section 4.

The third result will be the realizability interpretation of the theory of the axiomatic system $\mathbf{Par}(\mathsf{R})$, where we write simply $\mathbf{Par}(\mathsf{R})$ for $\mathbf{Par}_2(\mathsf{R})$. We also write $\mathbf{Par}(T)$ for $\mathbf{Par}_2(T)$. We shall define a realizability interpretation of the basic logic **L** into **L** itself, and we shall prove that $\mathbf{Par}(\mathsf{R})$ is self-realisable, that is, for each $\mathbf{Par}(T)$ in $\mathbf{Par}(\mathsf{R})$, there is a term R such that $\mathbf{Par}(\mathsf{R})$ proves that R realizes $\mathbf{Par}(T)$. As a consequence of this, for each theorem P of $\mathbf{Par}(\mathsf{R})$, there is a term R such that $\mathbf{Par}(\mathsf{R})$ proves that R realizes P. This result implies the term extraction property for the axiomatic system $\mathbf{Par}(\mathsf{R})$. It is given in Section 5.

In Section 2 of this paper, we shall first define a logical system **L**, and then use it as the basic logic for the axiomatic system of parametricity. The system **L** is a second order predicate logic on terms of the system F. It is slightly modified from the system of the basic logic of Plotkin and Abadi [8]. Their system and **L** are logically equivalent to each other, that is, there are interpretations of their system into **L**, and of **L** into their system, which preserve provabilities and logical meanings. But there are two main differences between our logic and their logic. One is that **L** has n-ary predicate variables for an arbitrary number n, while their system restricts predicate variables to be binary. This is not an essential difference, because both systems have pairing, so polyadic predicates can be encoded into binary predicates. The other difference is that our logic does not have equality as a primitive symbol; it is defined as Leibniz equality in our logic, while their logic has equality as a primitive symbol. This difference is used essentially in relativizing predicates in the proof of consistency. From another point of view, **L** is a subsystem of λP2 in the λ-cube which is studied by Barendregt [2], though **L** distinguishes types and formulae but λP2 does not distinguish them. The terms and types of **L** are those of System F. For example, $\forall x.\phi x \to \psi x$ is a formula, but not a type of **L**. But λP2 has $\Pi x.\phi x \to \psi x$ as a type, and $y^{\Pi x.\phi x \to \psi x}$ as a variable.

Continuing Section 2, we shall next define several notions in order to define the axiomatic system **Par** for parametricity. First, conformity for a type T will be defined by induction on the construction of T. Conformity will be a kind of logical relation suited to parametric polymorphism. We use the notation $\langle T\rangle\{\Phi_1/\gamma_1, \ldots, \Phi_i/\gamma_i\}$ to denote the notion of conformity for the type T. An outline of its definition is the following:

- $\langle\gamma\rangle\{\ldots, \Phi/\gamma, \ldots\} :\equiv \Phi$ for a type variable γ.
- $\langle T \to U\rangle\{\ldots\}xy$ iff for all v and w, if $\langle T\rangle\{\ldots\}vw$ then $\langle U\rangle\{\ldots\}(xv)(yw)$.
- $\langle\Pi\gamma.T\rangle\{\ldots\}xy$ iff $\langle T\rangle\{\phi/\gamma, \ldots\}(x\alpha)(y\beta)$ for all types α and β, and for all relations ϕ.

If the assignment $\{\ldots\}$ does not assign a predicate to a type variable γ , then the assignment is regarded as assigning equality to γ. We shall write $\langle T\rangle^=$ for $\langle T\rangle\{\}$. We shall say that x^T and y^T *conform* to each other when $\langle T\rangle^= xy$ holds. We shall say x^T is *self-conforming* when x conforms to x itself. We shall say $\phi^{T\to *}$ is *extensional* when ϕx is equivalent to ϕy if x and y conform to each other. Conformity of a type T will be a property that all the terms of the type T are self-conforming. Extensionality of a type T will be the property that all the predicates of predicate type $T \to *$ are extensional. Parametricity of T will be the conjunction of conformity and extensionality of T. We shall write $\mathbf{Par}(T)$ for parametricity of T. **Par** will be an axiomatic system, and its intuitive meaning will be the conjunction of $\mathbf{Par}(T)$ for all types T. The intuitive meaning of parametricity will be that x^T and y^T are equal to each other if and only if they conforms to each other. The axiomatic system **Par** is logically equivalent to the axiomatic system in [8]. As compared to [8] however, we divide the notion of parametricity into two notions: conformity and extensionality. The axioms in [8] only say that conformity and equality are equivalent to each other. The two notions, conformity and extensionality, are the essential points in relativizing predicates as well as Leibniz equality.

In Section 3, we shall show that the axiomatic system **Par** is consistent. That will be the first main result of this paper. We shall prove consistency by the method of relativization. Because equality is defined as Leibniz equality, an equation serves as a statement for predicates as well as a statement for terms. By using this point, we shall define the relativization. It will relativize terms by self-conformity, and relativize predicates by extensionality. The relativization will be sound and complete for the theory of **Par**, that is, a formula P is provable in the axiomatic system **Par** if and only if the relativized formula $rel(P)$ is provable in **L**.

In Section 4, we shall discuss polyadic parametricity, and give a partial answer to the problem of Plotkin and Abadi [8]. Our answer will be that the polyadic parametricities are equivalent to each other as long as the types are restricted to recursive types. That will be the second main result of this paper. Plotkin and Abadi [8] define n-ary parametricity: n-ary parametricity of a type T is that $\langle T\rangle^=_n x_1 \ldots x_n$ is equivalent to $x_1 = x_2 = \cdots = x_n$, where $\langle T\rangle^=_n$ is an n-ary conformity notation defined in a similar way to the binary. We write $\mathbf{Par}_n(T)$ for n-ary parametricity of a type T, and $\mathbf{Par}_n(\mathsf{R})$ for the set of axioms $\mathbf{Par}_n(T)$ for all the types T in R. The result is that the theories of $\mathbf{Par}_m(\mathsf{R})$ and of $\mathbf{Par}_n(\mathsf{R})$ are equal to each other for arbitrary numbers $m, n \geq 2$. In other words, for any T in R, the formula $\mathbf{Par}_n(T)$ is provable by $\mathbf{Par}_m(\mathsf{R})$.

In order to prove the result, we shall show that $\langle T\rangle^=_m$ and $\langle T\rangle^=_n$ are equivalent under the theory of $\mathbf{Par}_m(\mathsf{R})$. From this fact, we shall obtain that $\langle T\rangle^=_n$ is equivalent to equality, because $\mathbf{Par}_m(T)$ implies that $\langle T\rangle_m\{\}$ is equivalent to equality. But we cannot compare $\langle T\rangle^=_m$ and $\langle T\rangle^=_n$ directly, because $\langle T\rangle^=_m$ is an m-ary predicate and $\langle T\rangle^=_n$ is n-ary. Therefore we shall define filters $\mathbf{C}^m_n[\ ]$ and $\mathbf{D}^m_n[\ ]$ which translate an n-ary predicate into an m-ary predicate. The intuitive meaning of $\mathbf{C}^m_n[\Phi]$ will be an internal partial equivalence relation in the domain of Φ, and the intuitive meaning of $\mathbf{D}^m_n[\Phi]$ will be the symmetric transitive closure of Φ. The crucial lemma will be that $\mathbf{Par}_l(\mathsf{R})$ proves

$$\forall x_1 \ldots x_k.\mathbf{D}^k_m[\langle T\rangle_m\{\ldots\}]x_1 \ldots x_k \supset \mathbf{C}^k_n[\langle T\rangle_n\{\ldots\}]x_1 \ldots x_k$$

for arbitrary numbers $k, l, m, n \geq 2$. We shall have also that $\langle T\rangle^{=}_m$ implies $\mathbf{D}^m_m[\langle T\rangle^{=}_m]$ and $\mathbf{C}^n_n[\langle T\rangle^{=}_n]$ implies $\langle T\rangle^{=}_n$. Thus from this crucial lemma, we shall obtain the equivalence of $\langle T\rangle^{=}_m$ and $\langle T\rangle^{=}_n$.

The proof of the lemma will be done by induction on the construction of the type T in R. In the step for functional types $T \to U$, the symmetry and transitivity of $\mathbf{C}[\ ]$ and $\mathbf{D}[\ ]$ will do essential work. In the steps for $\mu\gamma.T$ and $\nu\gamma.T$ of the proof, we shall use the following fixed point lemma. The fixed point lemma will involve fixed point property, whose intuitive meaning will be that the conformity of fixed point types is equivalent to the fixed point predicate of conformity. The fixed point lemma will say that $\mathbf{Par}_m(\mathsf{R})$ proves the fixed point property for n-ary predicates for any arities m and n independent of each other. This lemma will be crucial in the induction steps for $\mu\gamma.T$ and $\nu\gamma.T$. This lemma appears in [8] only for $m = n$. This paper extends the lemma in [8].

In Section 5, we shall give a realizability interpretation for the theory of $\mathbf{Par}(\mathsf{R})$. First we shall define a realizability interpretation of the basic logic $\mathbf{L}$ into $\mathbf{L}$ itself. This interpretation will be sound and complete in the following sense. Soundness means that for each theorem P of $\mathbf{L}$, there is a term R such that $\mathbf{L}$ derives that R realizes P; completeness means that $\mathbf{L}$ proves that if R realizes P, then P holds. Next we shall show that the axiomatic system $\mathbf{Par}(\mathsf{R})$ is self-realizable, that is, there is a term R_T for each axiom $\mathbf{Par}(T) \in \mathbf{Par}(\mathsf{R})$ such that $\mathbf{Par}(\mathsf{R})$ proves that R_T realizes $\mathbf{Par}(T)$. In order to prove the self-realizability of $\mathbf{Par}(\mathsf{R})$, we shall define a notion $\langle T\rangle^{\models}$. The intuitive meaning of $\langle T\rangle^{\models}rxy$ is that r realizes $\langle T\rangle^{=}xy$. We shall show the crucial lemma which says that $\mathbf{Par}(\mathsf{R})$ proves that $\langle T\rangle^{=}_3xyz$ implies $\langle T\rangle^{\models}(K_Tx)yz$ for some term K_T, for each type $T \in \mathsf{R}$. The proof of this crucial lemma will be done by induction on construction of $T \in \mathsf{R}$.

2. Logic for Parametricity

In this section we define a logical system $\mathbf{L}$ and a set of axioms for parametricity $\mathbf{Par}$ on $\mathbf{L}$. In order to define $\mathbf{Par}$, we introduce the notion of conformity in advance.

2.1. The Basic Logic

First we define the basic logic $\mathbf{L}$ on which the theory of parametricity is founded. It is the second order logic on the second order typed lambda calculus, which is studied as System F by Girard [4].

Definition 2.1. Here we define the second order logic on the second order typed lambda calculus, and we call it $\mathbf{L}$.

The language of $\mathbf{L}$ is defined as follows:

– *Types* and *terms* are those of System F, that is:

$$T ::= \gamma \mid T \to T \mid \Pi\gamma.T \quad \text{and} \quad X ::= x^T \mid \lambda x^T.X \mid XX \mid \Lambda\gamma.X \mid XT$$

where the application is done only when the types of the two terms involved are suitable. All the variables are explicitly typed, thus all the subterms are typed. But we often omit the superscripts which indicate types.

– *Predicate types*, *predicates*, and *formulae* are constructed as:

Predicate types: $A ::= * \mid T \to A$
Predicates: $\Phi ::= \phi^A \mid (\lambda x^T.\Phi^A)^{T\to A} \mid (\Phi^{T\to A}X^T)^A \mid P^*$
Formulae: $P ::= \Phi^* \mid P \supset P \mid \forall\gamma.P \mid \forall x^T.P \mid \forall\phi^A.P$

All the subexpressions are explicitly typed as well as variables and predicate variables. But we often omit the superscripts which indicate types.
Formulae are the predicates of predicate type $*$. We use the letters $x, y, \ldots$ for variables, $X, Y, \ldots$ for terms, $\alpha, \beta, \gamma, \delta, \ldots$ for type variables, and $T, U, A, B, \ldots$ for types. A and B are also used for predicate types. $P, Q, \ldots$ are for formulae, $\phi, \psi, \ldots$ for predicate variables, and Φ, Ψ for predicates.

The inference rules are given below. In the expression "$E, \Gamma \vdash P$", E is an environment which declares types of variables such as $\{x : T, y : U, \ldots\}$, Γ is an assumption, which is a finite set of formulae, and P is a formula. An expression Γ, Δ stands for $\Gamma \cup \Delta$, and Γ, P stands for $\Gamma \cup \{P\}$, and so forth.

When X is a term or a predicate, $Env(X)$ is the least environment which is needed for constructing X. The notations $Env(X_1, X_2, \ldots)$ and $Env(\{X_1, X_2, \ldots\})$ stand for $\bigcup_i Env(X_i)$.

- Structural Rules:

$$\frac{}{E, \Gamma, P \vdash P}\,, \qquad \frac{E, \Gamma \vdash P}{E, \Gamma, Q \vdash P}\,, \qquad \frac{E, \Gamma \vdash P}{E, x : T, \Gamma \vdash P}$$

where $Env(P) \subset E$, $Env(Q) \subset E$ and $Env(\Gamma) \subset E$.

- Implication Introduction:

$$\frac{E, \Gamma, P \vdash Q}{E, \Gamma \vdash P \supset Q}$$

- Implication Elimination:

$$\frac{E, \Gamma \vdash P \supset Q \quad E, \Gamma \vdash P}{E, \Gamma \vdash Q}$$

- Term Quantification Introduction:

$$\frac{E, x : T, \Gamma \vdash P}{\Gamma \vdash \forall x.P}$$

where x does not occur free in Γ.

- Term Quantification Elimination:

$$\frac{E, \Gamma \vdash \forall x^T.P}{E, \Gamma \vdash P[X^T/x]}$$

where $Env(X) \subset E$.

- Predicate Quantification Introduction:

$$\frac{E, \Gamma \vdash P}{E, \Gamma \vdash \forall\phi.P}$$

where ϕ does not occur free in Γ.

- Predicate Quantification Elimination:

$$\frac{E, \Gamma \vdash \forall\phi^A.P}{E, \Gamma \vdash P[\Phi^A/\phi]}$$

where $Env(\Phi) \subset E$.

- Type Quantification Introduction:

$$\frac{E, \Gamma \vdash P}{E, \Gamma \vdash \forall\gamma.P}$$

where γ does not occur free in Γ, and moreover, γ does not occur free in the types of free variables, nor the predicate types of free predicate variables in P.

- Type Quantification Elimination:

$$\frac{E, \Gamma \vdash \forall\gamma.P}{E, \Gamma \vdash P[T/\gamma]}$$

- β-Conversion:

$$\frac{E, \Gamma \vdash P}{E, \Gamma \vdash Q}$$

where there is a one-step β-reduction from P to Q, or from Q to P; and $Env(Q) \subset E$. Both $(\lambda x.X)Y$ and $(\lambda x.\Phi)X$ are reducible.

If E or Γ is an infinite set, then $E, \Gamma \vdash P$ is defined to mean that there are finite subsets $E' \subset E$ and $\Gamma' \subset \Gamma$ such that $E', \Gamma' \vdash P$. If $\vdash P$ is derivable, then P is called a *theorem* of $\mathbf{L}$, and we write $\mathbf{L} \vdash P$. The set of all theorems of $\mathbf{L}$ is called the *theory* of $\mathbf{L}$. Let $\mathbf{A}$ be a set of formulae regarded as axioms. If P is derivable from $\mathbf{A}$, then P is called a *theorem* of $\mathbf{A}$, and the set of all formulae P such that $\mathbf{A} \vdash P$ is called the *theory* of $\mathbf{A}$.

The system $\mathbf{L}$ is logically equivalent to the basic logic in [8], that is, there are translations of the basic logic in [8] into $\mathbf{L}$ and of $\mathbf{L}$ into the basic logic in [8], which preserve provabilities and logical meanings. The translation of the basic logic in [8] into $\mathbf{L}$ is the identity function, because all the formulae and proofs of the basic logic in [8] are also formulae and proofs of $\mathbf{L}$. The inverse is defined so that it translates all polyadic predicates into binary predicates. For example, a ternary predicate $\Phi^{A \to B \to C \to *}$ is translated into a binary predicate $\lambda x^A y^{B \times C}.\Phi x y' y'' \wedge y = \boldsymbol{\pi} y' y''$, where $\wedge$ and $=$ are defined in Definition 2.4, and $\times$ and $\boldsymbol{\pi}$ are defined in Section 2.4.

2.2. Preliminaries

Here are some important propositions and definitions for $\mathbf{L}$. The propositions and lemmata say that $\mathbf{L}$ is situated between $\lambda 2$ and λP2 in the Lambda-Cube, which is studied by Barendregt [2].

Lemma 2.2. $\mathbf{L}$ is consistent.

Proof:
By forgetting all the terms and all the inferences of term quantification, a derivation of $\mathbf{L}$ is translated into a derivation of second order propositional logic. If we have a proof of $\forall\phi.\phi$ in $\mathbf{L}$, then we also have a proof of $\forall\phi.\phi$ in second order propositional logic. Hence, the consistency of $\mathbf{L}$ is reduced to the consistency of second order propositional logic. □

This proof uses a translation which translates derivations of $\mathbf{L}$ onto those of second order propositional logic. This translation appears in [5].

Proposition 2.3. $\mathbf{L}$ is a subsystem of λP2 which is closed under β-reduction.

This proposition means that we can embed $\mathbf{L}$ into λP2 by regarding types and formulae of $\mathbf{L}$ as types of λP2, and terms and derivations of $\mathbf{L}$ as terms of λP2. Let D be a derivation of $\mathbf{L}$. If D is regarded as a term of λP2, and reduced into a term D' of λP2, then D' is again a derivation of $\mathbf{L}$. Thus the normalizability of derivations of $\mathbf{L}$ follows from that of λP2. By normalizing derivations, we get the disjunction property and the term extracting property of $\mathbf{L}$. Later we shall prove them again by using the realizability interpretation.

Note that the converse of this translation is not always successful. The logic $\mathbf{L}$ does not derive $x : \Pi\gamma.\gamma \vdash \forall\phi.\phi$, although λP2 derives its interpretation, which is $x : \Pi\gamma : *.\gamma \vdash x : \Pi\gamma : *.\gamma$.

Now we define some notations.

Definition 2.4.

$$\begin{aligned}
&\mathbf{I}_T :\equiv \lambda x^T.x,\\
&X^{U\to V} \circ Y^{T \to U} :\equiv \lambda x^T.X(Yx),\\
&\bot :\equiv \forall\phi^*.\phi,\\
&P \wedge Q :\equiv \forall\phi^*.(P{\supset}Q{\supset}\phi){\supset}\phi,\\
&P \vee Q :\equiv \forall\phi^*.(P{\supset}\phi){\supset}(Q{\supset}\phi){\supset}\phi,\\
&\exists x.P :\equiv \forall\phi^*.(\forall x.P{\supset}\phi){\supset}\phi,\\
&P \supset\subset Q :\equiv (P{\supset}Q) \wedge (Q{\supset}P),\\
&X =_T Y :\equiv \forall\phi^{T\to *}.\phi X \supset \phi Y,\\
&\mathcal{E}q_T :\equiv \lambda xy.(x =_T y),
\end{aligned}$$

$\bot^c :\equiv \forall\gamma.\forall x^\gamma y^\gamma.x =_\gamma y,$
$X^{T\to U} \simeq Y^{T\to U} :\equiv \forall x^T.Xx =_U Yx,$
$\Phi \sqsubseteq \Psi :\equiv \forall x_1 \dots x_n.\Phi x_1 \dots x_n \supset \Psi x_1 \dots x_n,$
$\Phi \cong \Psi :\equiv \Phi \sqsubseteq \Psi \wedge \Psi \sqsubseteq \Phi.$

We often omit the type suffixes, namely, T in $\mathbf{I}_T$, $=_T$ and $\mathcal{E}q_T$.

The formula $\bot^c$ collapses all terms of each type, thus $\bot^c$ serves the contradiction for the theory of calculation, while $\bot$ serves the contradiction for logic.

In this paper, $\equiv$ denotes identity of expressions and $=_\beta$ denotes β-equality of expressions. We write $\{\dots\}$ for assignments and $[\dots]$ for substitutions.

Lemma 2.5. $\mathbf{L}$ does not collapse calculation, that is, $\mathbf{L} \nvdash \bot^c$.

Proof:
If $\mathbf{L} \vdash \bot^c$, then λP2 has some closed term which is of type $\bot^c$. But there are no closed normal terms of type $\bot^c$ in λP2. This contradicts the normalizability of λP2. □

2.3. Conformity, Extensionality and Parametricity

Now we define the conformity notation for $\mathbf{L}$.

Definition 2.6. Let T be a type and $\gamma_1, \dots, \gamma_n$ be type variables. Let $\Phi_1, \dots, \Phi_n$ be binary predicates whose predicate types are $A_1 \to B_1 \to *, \dots, A_n \to B_n \to *$ respectively. Let T_1, T_2 be types such that $T_1 \equiv T[A_1/\gamma_1, \dots, A_n/\gamma_n]$ and $T_2 \equiv T[B_1/\gamma_1, \dots, B_n/\gamma_n]$.

Then $\langle T\rangle\{\Phi_1/\gamma_1, \dots, \Phi_n/\gamma_n\}$ is a binary predicate, whose predicate type is $T_1 \to T_2 \to *$. It is called the *conformity notation*, and is defined by the induction on T given below. In this definition, $\Theta :\equiv \{\Phi_1/\gamma_1, \dots, \Phi_n/\gamma_n\}$.

- For a type variable γ:
  $\langle\gamma\rangle\{\Theta\} :\equiv \Phi$ if Θ assigns Φ to γ.
  $\langle\gamma\rangle\{\Theta\} :\equiv \mathcal{E}q_\gamma$ if Θ does not assign a predicate to γ.
- $\langle T \to U\rangle\{\Theta\} :\equiv \lambda xy.\forall vw.\langle T\rangle\{\Theta\}vw \supset \langle U\rangle\{\Theta\}(xv)(yw).$
- $\langle\Pi\gamma.T\rangle\{\Theta\} :\equiv \lambda xy.\forall\alpha\beta.\forall\phi^{\alpha\to\beta\to *}.\langle T\rangle\{\phi/\gamma, \Theta\}(x\alpha)(y\beta).$

For example:

$$\langle\Pi\gamma.\gamma \to \gamma\rangle\{\ \}XY =_\beta \forall\alpha\beta.\forall\phi^{\alpha\to\beta\to *}.\forall v^\alpha w^\beta.\phi xy \supset \phi(X\alpha v)(Y\beta w).$$

Definition 2.7. The notations $\langle\ \rangle^=$, $\mathbf{Conf}(\ )$, $\mathbf{Ext}(\ )$ and $\mathbf{Par}(\ )$ are defined as below:

- $\langle T\rangle^= :\equiv \langle T\rangle\{\}$.
- For a type T and a term X^T:
  $\mathbf{Conf}_T(X) :\equiv \langle T\rangle^= XX.$

We call this the *self-conformity* of X. When it holds, we say that X is *self-conforming.*

- $\mathbf{Conf}(T) :\equiv \forall x^T.\mathbf{Conf}_T(x)$. We call this the *conformity* of T.
- For a type T and a predicate $\Phi^{T\to *}$:
  $\mathbf{Ext}_T(\Phi) :\equiv \forall x^T y^T.\langle T\rangle^= xy \supset \Phi x \supset \Phi y.$

We call this the *extensionality* of Φ. When it holds, we say that Φ is *extensional.*

- $\mathbf{Ext}(T) :\equiv \forall\phi^{T\to *}.\mathbf{Ext}_T(\phi)$. We call this the *extensionality* of T.
- $\mathbf{Par}(T) :\equiv \mathbf{Conf}(T) \wedge \mathbf{Ext}(T)$, which is called the *parametricity* of T.

In $\mathbf{L}$, $\mathbf{Conf}(T)$ is equivalent to: $\forall x^T y^T.x = y \supset \langle T\rangle^= xy$.
$\mathbf{Ext}(T)$ is equivalent to: $\forall x^T y^T.\langle T\rangle^= xy \supset x = y$.
And $\mathbf{Par}(T)$ is equivalent to: $\forall x^T y^T.x = y \supset\subset \langle T\rangle^= xy$.

Definition 2.8. $\mathbf{Par}$ is the set consisting of the axioms $\forall\gamma_1 \dots \gamma_n.\mathbf{Par}(T)$ where $\gamma_1, \dots, \gamma_n$ are all the free type variables of T, for all types T.

The theory of $\mathbf{Par}$ is logically equivalent to the system of [8]. In the next subsection we will mention that the parametricities infer the universalities of universal types, where the universalities are read in the sense of categorical theory.

2.4. Recursive Types

Here are some data structures which are very often used. The theory of parametricity proves the universalities of these structures.

Definition 2.9. (Unit Type, Pair Type, Sum Type)
Unit type: $\mathbf{1} :\equiv \Pi\gamma.\gamma \to \gamma$.
Pair type: $T \times U :\equiv \Pi\gamma.(T \to U \to \gamma) \to \gamma$.
Sum type: $T + U :\equiv \Pi\gamma.(T \to \gamma) \to (U \to \gamma) \to \gamma$.

We call $T \to U$ a functional type. It is well-known that the theory of parametricity proves the universalities of these structures [6, 7].x1 For example:

Proposition 2.10. $\mathbf{Par}(\gamma \times \delta) \vdash \forall x^{\gamma\times\delta}.x = \boldsymbol{\pi}(\mathbf{pl}x)(\mathbf{pr}x)$,
where $\boldsymbol{\pi}$ is the standard pairing, and **pl** and **pr** are the standard projections.

Definition 2.11. (Fixed Point Type)
$\mu\gamma.T :\equiv \Pi\gamma.(T \to \gamma) \to \gamma$,
$\nu\gamma.T :\equiv \Pi\delta.(\Pi\gamma.(\gamma \to T) \to \gamma \to \delta) \to \delta$,
where γ occurs only positively in T. The type $\mu\gamma.T$ is called the *least fixed point type*, and the type $\nu\gamma.T$ is called the *greatest fixed point type*.

It is well-known that $\mu\gamma.T$ can encode recursive structures, and $\nu\gamma.T$ can encode lazy structures. We discuss the properties of fixed point types in Section 4.3.

Definition 2.12. R is the set of all the types which are constructed from $\mathbf{1}$ and type variables by the pair type $\times$, the sum type $+$, the functional type $\to$ and the fixed point types μ and ν. Types in R are called *recursive types*. $\mathbf{Par}(\mathsf{R})$ is the set consisting of axioms $\mathbf{Par}(T)$ for all types $T \in \mathsf{R}$.

3. Consistency of the Parametricity Axioms

In this section we prove the consistency of **Par**. It is done by the method of relativization, and is one of main results in this paper.

3.1. Relativized Parametricity

Definition 3.1. Here we define the notations $\langle T\rangle_{rel}\{\ldots\}$ and $\langle T\rangle_{rel}^{=}$, which are called *relativized conformity notations*. They are defined by mutual induction on T. The definition of $\langle\ \rangle_{rel}\{\ \}$ is as below:

- For a type variable γ:
  $\langle\gamma\rangle_{rel}\{\Theta\} :\equiv \Phi$ if Θ assigns Φ to γ.
  $\langle\gamma\rangle_{rel}\{\Theta\} :\equiv \mathcal{E}q_\gamma$ if Θ does not assign a predicate to γ.
- $\langle T \to U\rangle_{rel}\{\Theta\} :\equiv$
  $\lambda xy.\forall vw.\langle T\rangle_{rel}^{=} vv \supset \langle T\rangle_{rel}^{=} ww \supset \langle T\rangle_{rel}\{\Theta\}vw \supset \langle U\rangle_{rel}\{\Theta\}(xv)(yw)$.
- $\langle\Pi\gamma.T\rangle_{rel}\{\Theta\} :\equiv \lambda xy.\forall\alpha\beta\forall\phi.\langle T\rangle_{rel}\{\phi/\gamma,\Theta\}(x\alpha)(y\beta)$.

The definition of $\langle T\rangle_{rel}^{=}$ is: $\langle T\rangle_{rel}^{=} :\equiv \langle T\rangle_{rel}\{\ \}$.

The definition of $\langle-\rangle_{rel}\{\ldots\}$ is quite similar to that of $\langle-\rangle\{\ldots\}$. The difference between them is that the quantifier is relativized in the definition of $\langle T \to U\rangle_{rel}\{\ldots\}$.

Definition 3.2. Here are the definitions of relativized self-conformity and relativized extensionality.

$\mathbf{Conf}^{rel}_T(X) :\equiv \langle T\rangle^{=}_{rel}\, XX,$

$$\begin{aligned}\mathbf{Ext}^{rel}_{T_1 \to \dots \to T_n \to *}(\Phi) :\equiv \forall x_1^{T_1} \dots x_n^{T_n} y_1^{T_1} \dots y_n^{T_n}.&\\ &\mathbf{Conf}^{rel}(x_1) \supset \dots \supset \mathbf{Conf}^{rel}(x_n) \supset\\ &\mathbf{Conf}^{rel}(y_1) \supset \dots \supset \mathbf{Conf}^{rel}(y_n) \supset\\ &\langle T_1\rangle^{=}_{rel}\, x_1 y_1 \supset \dots \supset \langle T_n\rangle^{=}_{rel}\, x_n y_n \supset \Phi x_1 \dots x_n \supset \Phi y_1 \dots y_n.\end{aligned}$$

Now we define parametricity relativization. This relativization is a translation from formulae into formulae which rewrites quantifiers. We define the translation for predicates for convenience of definition.

Definition 3.3. For a predicate Φ, a predicate $rel(\Phi)$ is defined by induction on Φ as follows:

$rel(\phi) :\equiv \phi,$ for a predicate variable ϕ.
$rel(\Phi X) :\equiv rel(\Phi)X,$ for a predicate ΦX.
$rel(\lambda x.\Phi) :\equiv \lambda x.rel(\Phi),$ for a predicate $\lambda x.\Phi$.
$rel(P \supset Q) :\equiv rel(P) \supset rel(Q).$
$rel(\forall x^T.P) :\equiv \forall x^T.\mathbf{Conf}^{rel}_T(x) \supset rel(P).$
$rel(\forall \phi^A.P) :\equiv \forall \phi^A.\mathbf{Ext}^{rel}_A(\phi) \supset rel(P).$
$rel(\forall \gamma.P) :\equiv \forall \gamma.rel(P).$

The intuitive meaning of this relativization is that terms are relativized by self-conformity, and predicates are relativized by extensionality. Leibniz equality and extensionality play essential roles in relativizing a formula $x = y$. The notion of Leibniz equality is that equality is regarded as the relation between two objects which behave in the same way for any properties. In the formal notation, Leibniz equality of x and y is written as $\forall \phi.\phi x \supset \phi y$. Our relativization relativizes this $\forall \phi$ by extensionality.

3.2. Soundness and Completeness

This subsection is aimed at proving the following theorem.

Theorem 3.4. The above parametricity relativization is sound and complete for the theory of **Par**, that is, $\mathbf{Par} \vdash P$ iff $\mathbf{L} \vdash rel(P)$ for every closed formula P.

To prove this, we need to prove some lemmata. Theorem 3.4 is proved from the soundness and the completeness of the relativization. We shall obtain the consistency of **Par** from the soundness. First we show the soundness.

Lemma 3.5. Let X be a term of type T, and let $x_1^{T_1} \dots x_n^{T_n}$ be all the free variables in X. Then **L** derives:

$$Env(X), Env(Y), \langle T_1\rangle^{=}_{rel}\, x_1 y_1, \dots, \langle T_n\rangle^{=}_{rel}\, x_n y_n \vdash \langle T\rangle^{=}_{rel}\, XY,$$

where $Y :\equiv X[y_1/x_1, \dots, y_n/x_n]$.

Proof:
By induction on the construction of X. $\square$

Lemma 3.6. Let Φ be a predicate, and let $x_1, \dots, x_m, \phi_1, \dots, \phi_n$ be all the variables and predicate variables in Φ. Then **L** derives:

$$Env(\Phi), \mathbf{Conf}^{rel}(x_1), \dots, \mathbf{Conf}^{rel}(x_m), \mathbf{Ext}^{rel}(\phi_1), \dots, \mathbf{Ext}^{rel}(\phi_n) \vdash \mathbf{Ext}^{rel}(rel(\Phi)).$$

Proof:
By induction on the construction of Φ. $\Box$

Lemma 3.7. (Soundness of Relativization) Let $P, Q_1, \ldots, Q_n$ be formulae, and let $x_1 \ldots x_l, \phi_1, \ldots, \phi_m$ be all the variables and predicate variables in $P, Q_1, \ldots, Q_n$. If

$$Env(P, Q_1, \ldots, Q_n), \mathbf{Par}, Q_1, \ldots, Q_n \vdash P,$$

then $\mathbf{L}$ derives:

$$Env(P), \mathbf{Conf}^{rel}(x_1), \ldots, \mathbf{Conf}^{rel}(x_l), \\ \mathbf{Ext}^{rel}(\phi_1), \ldots, \mathbf{Ext}^{rel}(\phi_m), rel(Q_1), \ldots, rel(Q_n) \vdash rel(P).$$

Proof:
By induction on the derivation of $Env(P), \mathbf{Par}, Q_1, \ldots, Q_n \vdash P$. In the induction steps for the rules of quantification eliminations, we note that the following property on substitution holds.

$$rel(P[X/x]) \equiv rel(P)[X/x],$$
$$rel(P[\Phi/\phi]) \equiv rel(P)[rel(\Phi)/\phi],$$
$$rel(P[T/\gamma]) \equiv rel(P)[T/\gamma].$$

$\Box$

Next we show completeness.

Proposition 3.8. $\mathbf{Par} \vdash \forall x^T.\mathbf{Conf}^{rel}_T(x)$ and $\mathbf{Par} \vdash \forall \phi^A.\mathbf{Ext}^{rel}_A(\phi)$.

Proof:
By induction on the constructions of T and A. $\Box$

Proposition 3.9. $Env(P), \mathbf{Par} \vdash P \supset\subset rel(P)$ for any formula P.

Proof:
By induction on the construction of P. $\Box$

Lemma 3.10. (Completeness of Relativization) Let P be a formula, and let $x_1, \ldots, x_m$, $\phi_1 \ldots \phi_n$ be all the variables and the predicate variables in P. If $\mathbf{L}$ derives:

$$Env(P), \mathbf{Conf}^{rel}(x_1), \cdots \mathbf{Conf}^{rel}(x_m), \mathbf{Ext}^{rel}(\phi_1), \cdots \mathbf{Ext}^{rel}(\phi_n) \vdash rel(P),$$

then $Env(P), \mathbf{Par} \vdash P$.

Proof:
By Proposition 3.8 and 3.9. $\Box$

Corollary 3.11. (Consistency of Par) $\mathbf{Par}$ is consistent.

Proof:
From soundness (Lemma 3.7), if $\mathbf{Par} \vdash \bot$, then $\mathbf{L} \vdash rel(\bot)$. But $rel(\bot) \equiv \forall \phi.(\phi \supset \phi) \supset \phi$, which is equivalent to $\bot$. Thus, if $\mathbf{Par}$ is inconsistent, then $\mathbf{L}$ is inconsistent. This contradicts Lemma 2.2. $\Box$

Corollary 3.12. $\mathbf{Par}$ does not collapse calculation, that is, $\mathbf{Par} \not\vdash \bot^c$.

Proof:
The translation $rel(\bot^c)$ is equivalent to $\bot^c$, but $\mathbf{L} \not\vdash \bot^c$ by Lemma 2.5. $\Box$

4. Polyadic Parametricity

We discussed parametricity only for binary relations until the previous section. Plotkin and Abadi [8] give the definition of polyadic parametricity. In this section we discuss the equivalences among the theories of polyadic parametricities.

4.1. Definitions, Conjecture and Theorem

The definition of polyadic parametricity is similar to the binary one. It is essentially equivalent to that in [8] just the same as for binary relations.

Definition 4.1. $\mathcal{E}q^n_T :\equiv \lambda x_1 \dots x_n.(x_1 =_T x_2 \wedge x_1 =_T x_3 \wedge \dots \wedge x_1 =_T x_n)$.

Definition 4.2. Let T be a type, n be a number ≥ 2 and $\Phi_1, \dots, \Phi_m$ be n-ary predicate variables, where Φ_i is of type $U_{i,1} \to \dots \to U_{i,n} \to *$.

Then $\langle T \rangle_n \{\Phi_1/\gamma_1, \dots, \Phi_m/\gamma_m\}$ is a predicate of type $T_1 \to \dots \to T_n \to *$, where $T_i :\equiv T[U_{1,i}/\gamma_1, \dots, U_{m,i}/\gamma_m]$. It is defined by induction on T as below.

– For a type variable:

$\langle \gamma \rangle_n \{\Theta\} :\equiv \Phi$ if Θ assigns Φ to γ.
$\langle \gamma \rangle_n \{\Theta\} :\equiv \mathcal{E}q^n_\gamma$ if Θ does not assign a predicate to γ.

– $\langle T \to U \rangle_n \{\Theta\} :\equiv \lambda x_1 \dots x_n.\forall y_1 \dots y_n.\langle T \rangle_n \{\Theta\} y_1 \dots y_n \supset \langle U \rangle_n \{\Theta\}(x_1 y_1) \dots (x_n y_n)$.

– $\langle \Pi\gamma.T \rangle_n \{\Theta\} :\equiv \lambda x_1 \dots x_n.\forall \alpha_1 \dots \alpha_n.\forall \phi^{\alpha_1 \to \dots \to \alpha_n \to *}.\langle T \rangle_n \{\phi/\gamma, \Theta\}(x_1 \alpha_1) \dots (x_n \alpha_n)$.

Proposition 4.3. $\langle T \rangle_n \{\langle U \rangle_n \{\Theta\}/\gamma, \Theta\} \equiv \langle T[U/\gamma] \rangle_n \{\Theta\}$

Proof:
By induction on T. □

The notations of $\langle T \rangle^=_n$, $\mathbf{Par}_n(T)$, $\mathbf{Par}_n$ and $\mathbf{Par}_n(\mathsf{R})$ are defined so that they denote the similar meanings to those for the binary case.

Definition 4.4. $\langle T \rangle^=_n$ and $\mathbf{Par}_n(T)$ are defined as:

$$\langle T \rangle^=_n :\equiv \langle T \rangle_n \{\},$$

$$\mathbf{Par}_n(T) :\equiv \forall x^T_1 \dots x^T_n.\mathcal{E}q^n x_1 \dots x_n \supset\subset \langle T \rangle^=_n x_1 \dots x_n.$$

$\mathbf{Par}_n$ is the set consisting of the axioms $\forall \gamma_1 \dots \gamma_n.\mathbf{Par}_n(T)$ where $\gamma_1, \dots, \gamma_n$ are all the free type variables of T, for all types T. $\mathbf{Par}_n(\mathsf{R})$ is the set consisting of axioms $\mathbf{Par}_n(T)$ for all types $T \in \mathsf{R}$.

Lemma 4.5. $\mathbf{Par}_n$ is consistent. Moreover, $\bigcup\limits_{m \leq n} \mathbf{Par}_m$ is consistent for any n. Therefore, $\bigcup\limits_n \mathbf{Par}_n$ is consistent.

Proof:
Similarly to the case for binary parametricity, there is a suitable relativization which is sound and complete for the theory of $\bigcup\limits_{m \leq n} \mathbf{Par}_m$. We shall give only a rough explanation of it.

For $2 \leq m \leq n$, the notations $\langle T \rangle^{rel \leq n}_m \{\Theta\}$, $\langle T \rangle^{rel \leq n}_m$, and $\mathbf{Conf}^{rel \leq n}_(X)$ is mutually defined as below.

– For a type variable:

$\langle \gamma \rangle^{rel \leq n}_m \{\Theta\} :\equiv \Phi$ if Θ assigns Φ to γ.
$\langle \gamma \rangle^{rel \leq n}_m \{\Theta\} :\equiv \mathcal{E}q^m_\gamma$ if Θ does not assign a predicate to γ.

– $\langle T \to U \rangle^{rel \leq n}_m \{\Theta\} :\equiv$

$$\lambda x_1 \dots x_m.\forall y_1 \dots y_m.\mathbf{Conf}^{rel \leq n}(y_1) \supset \cdots \mathbf{Conf}^{rel \leq n}(y_m) \supset \langle T \rangle_m^{rel \leq n}\{\Theta\} y_1 \dots y_m \supset \langle U \rangle_m^{rel \leq n}\{\Theta\}(x_1 y_1) \dots (x_m y_m).$$

- $\langle \Pi\gamma.T \rangle_m^{rel \leq n}\{\Theta\} :\equiv$
  $\lambda x_1 \dots x_m.\forall \alpha_1 \dots \alpha_m.\forall \phi^{\alpha_1 \to \cdots \to \alpha_n \to *}.\langle T \rangle_m^{rel \leq n}\{\phi/\gamma, \Theta\}(x_1 \alpha_1) \dots (x_m \alpha_m)$,
- $\langle T \rangle_m^{rel \leq n} :\equiv \langle T \rangle_m^{rel \leq n}\{\}$,
- $\mathbf{Conf}_T^{rel \leq n}(X) :\equiv \langle T \rangle_2^{rel \leq n} XX \wedge \langle T \rangle_3^{rel \leq n} XXX \wedge \cdots \wedge \langle T \rangle_n^{rel \leq n} X \dots X$.

The notations $\mathcal{E}q_T^{rel \leq n}$ and $\mathbf{Ext}_A^{rel \leq n}(\Phi)$ are defined as below.

- $\mathcal{E}q_T^{rel \leq n} :\equiv \lambda x^T y^T. \bigvee_{2 \leq m \leq n} \exists z_1 \dots z_m.\langle T \rangle_m^{rel \leq n} z_1 \dots z_m \wedge (\bigvee_{i \leq m} x = z_i) \wedge (\bigvee_{j \leq m} y = z_j)$
- $\mathbf{Ext}_{T_1 \to \cdots \to T_m \to *}^{rel \leq n}(\Phi) :\equiv$
  $\forall x_1^{T_1} \dots x_m^{T_m} y_1^{T_1} \dots y_m^{T_m}.\mathcal{E}q_{T_1}^{rel \leq n} x_1 y_1 \supset \cdots \supset \mathcal{E}q_{T_m}^{rel \leq n} x_m y_m \supset \Phi x_1 \dots x_m \supset \Phi y_1 \dots y_m$.

The relativization $rel_{\leq n}(P)$ translates the quantifications as follows:

$$rel_{\leq n}(\forall x.P) \equiv \forall x.\mathbf{Conf}^{rel \leq n}(x) \supset rel_{\leq n}(P),$$
$$rel_{\leq n}(\forall \phi.P) \equiv \forall \phi.\mathbf{Ext}^{rel \leq n}(\phi) \supset rel_{\leq n}(P).$$

Then this relativization $rel_{\leq n}()$ is sound and complete for the theory of $\bigcup_{m \leq n} \mathbf{Par}_m$, and $rel_{\leq n}(\perp)$ and $rel_{\leq n}(\perp)^c$ are equivalent to $\perp$ and $\perp^c$, respectively. $\square$

We make this conjecture for the polyadic parametricity: $\mathbf{Par}_m \vdash \mathbf{Par}_n(T)$ for all types T and all $m, n \geq 2$. This conjecture says that all the $\mathbf{Par}_n$'s are equivalent to each other for arbitrary numbers $n \geq 2$. However, we have only the following theorem yet.

Theorem 4.6. $\mathbf{Par}_m(\mathsf{R}) \vdash \mathbf{Par}_n(T)$ for all types T in R and all $m, n \geq 2$.

This theorem says that all of $\mathbf{Par}_n(\mathsf{R})$'s are equivalent to each other for arbitrary numbers $n \geq 2$. We will spend the remaining part of Section 4 in showing the proof of it.

4.2. Functor

In order to prove Theorem 4.6. we have to define the notion of functor and some notations which denote terms in the sense of Definition 2.1.

Definition 4.7. When a type variable γ occurs only positively in a type T, then the operation which makes $T[U/\gamma]$ from U is called a *functor*. When a type variable γ occurs only negatively in a type T, then the operation which makes $T[U/\gamma]$ from U is called a *contravariant functor*.

Suppose that $C[\,]$ is a context which makes a type $C[T]$ from a type T, and that a functor or contravariant functor F makes $C[T]$ from T. Then $C[-]$ denotes the functor or contravariant functor F.

Definition 4.8. A functor or a contravariant functor F is said to be in R if $F\gamma \in \mathsf{R}$.

Remember that the notations $\mathbf{I}$ and $\circ$ are defined in Definition 2.4 as: $\mathbf{I}_T :\equiv \lambda x^T.x$, $X \circ Y :\equiv \lambda x.X(Yx)$.

Definition 4.9. We define the operation of functors and contravariant functors for some terms. Let X be a term of type $T \to U$, and F be a functor, or a contravariant functor, then $\langle X \rangle_F$ is a term of type $FT \to FU$ if F is a functor, or of type $FU \to FT$ if F is a contravariant functor, and is defined as follows.

If γ does not occur in $F\gamma$, then $\langle X \rangle_F :\equiv \mathbf{I}_{F\gamma}$.

Otherwise:

- When $F\gamma \equiv \gamma$, then $\langle X \rangle_F :\equiv X$.
- When $F\gamma \equiv G\gamma \to H\gamma$, then $\langle X \rangle_F :\equiv \lambda x.\langle X \rangle_H \circ x \circ \langle X \rangle_G$.
- When $F\gamma \equiv \Pi\delta.C[\gamma, \delta]$, then $\langle X \rangle_F :\equiv \Lambda\delta.\langle X \rangle_{C[-,\delta]}$.

Proposition 4.10. (Composability) $\langle X\rangle_F \circ \langle Y\rangle_F =_\beta \langle X \circ Y\rangle_F$ for a functor F, and $\langle X\rangle_G \circ \langle Y\rangle_G =_\beta \langle Y \circ X\rangle_G$ for a contravariant functor G.

Proof:
By induction on F and G. $\square$

Definition 4.11. We define substitution of types in functors. Let F be a functor and θ be a substitution of types such as $[U_1/\delta_1, \ldots, U_n/\delta_n]$. We can take γ as it is a fresh type variable, that is, γ does not occur free in $U_1, \ldots, U_n$, and γ is different from all of $\delta_1, \ldots, \delta_n$. Then $F\theta$ is a functor which makes $F\theta T \equiv (F\gamma)[\theta, T/\gamma]$ from a type T. Substitution of types for contravariant functors is defined similarly.

These identities hold: $FT\theta \equiv F\theta(T\theta)$, $(\langle X\rangle_F)\theta \equiv \langle X\theta\rangle_{F\theta}$, and $(\mu\gamma.F\gamma)\theta \equiv \mu\gamma.F\theta\gamma$.

Definition 4.12. Let F be a functor or a contravariant functor, Θ be an assignment of n-ary predicates for type variables, and Φ be an n-ary predicate. Then: $F^\Theta\Phi :\equiv \langle F\gamma\rangle_n\{\Phi/\gamma, \Theta\}$.

By using this notation, we write the substitution of conformity asserted in Proposition 4.3 applied to a functor as follows.

Proposition 4.13. (Substitution Lemma) $F^\Theta(\langle T\rangle_n\{\Theta\}) \equiv \langle FT\rangle_n\{\Theta\}$

Definition 4.14. Let Φ be a predicate of predicate type $T_1 \to \cdots \to T_n \to *$, and X be a term of type $U \to T_i$. Then,

$$\Phi \circ_i X :\equiv \lambda x_1^{T_1} \ldots x_{i-1}^{T_{i-1}} y^U x_{i+1}^{T_{i-1}} \ldots x_n^{T_n}.\Phi x_1 \ldots x_{i-1}(Xy)x_{i+1} \ldots x_n.$$

If Θ be an assignment of predicates for type variables such as $\{\Phi_1/\gamma_1, \ldots, \Phi_n/\gamma_n\}$, we use the notation $Env(\Theta) :\equiv Env(\Phi_1, \ldots, \Phi_n)$.

In the following propositions, $\sqsubseteq$ and $\cong$ are defined as in Definition 2.4 as

$$\Phi \sqsubseteq \Psi :\equiv \forall x_1 \ldots x_n.\Phi x_1 \ldots x_n \supset \Psi x_1 \ldots x_n, \quad \text{and} \quad \Phi \cong \Psi :\equiv \Phi \sqsubseteq \Psi \wedge \Psi \sqsubseteq \Phi.$$

Proposition 4.15. (Functor Lemma) Let i be a number such that $1 \le i \le n$. Let Θ be an assignment of predicates to type variables $\{\Phi_1/\gamma_1, \ldots, \Phi_m/\gamma i_m\}$ where Φ_j is of predicate type $T_{j,1} \to \cdots \to T_{j,n} \to *$ for $j = 1, \ldots, n$, and let θ be a substitution of types $[T_{1,i}/\gamma_1, \ldots, T_{n,i}/\gamma_n]$. If F is a functor and G is a contravariant functor, then $\mathbf{L}$ derives the following:

$$Env(\Phi, \Psi, \Theta, X), \Phi \sqsubseteq \Psi \circ_i X \ \vdash \ F^\Theta\Phi \sqsubseteq F^\Theta\Psi \circ_i \langle X\rangle_{F\theta},$$
$$Env(\Phi, \Psi, \Theta, X), \Phi \sqsubseteq \Psi \circ_i X \ \vdash \ G^\Theta\Psi \sqsubseteq G^\Theta\Phi \circ_i \langle X\rangle_{G\theta}.$$

Proof:
By mutual induction on F and G. $\square$

Remember that the notation $\simeq$ is defined in Definition 2.4 as: $X^{T\to U} \simeq Y^{T\to U} :\equiv \forall z^T.Xz = Yz$.

Corollary 4.16. (Identity Expansion Lemma) For a functor or a contravariant functor F, it holds that $\mathbf{Par}_n(FT) \vdash \mathbf{I}_{FT} \simeq \langle \mathbf{I}_T\rangle_F$.

Proof:
First we have $\vdash \langle T\rangle_n^= \cong \langle T\rangle_n^= \circ_1 \mathbf{I}_T$ by β-conversion.

Then we have

$$F^{\{\}}(\langle T\rangle_n^=) \cong F^{\{\}}(\langle T\rangle_n^= \circ_1 \mathbf{I}_T).$$

Next, by the Substitution Lemma (Proposition 4.13) we have $F^{\{\}}(\langle T\rangle_n^=) \cong \langle FT\rangle_n^=$. Therefore we have

$$\langle FT\rangle_n^= \cong F^{\{\}}(\langle T\rangle_n^= \circ_1 \mathbf{I}_T).$$

Thus, by Functor Lemma (Proposition 4.15) we have $\langle FT\rangle_n^= \sqsubseteq F^{\{\}}(\langle T\rangle_n^=) \circ_1 \langle \mathbf{I}_T\rangle_F$, which is

$$\langle FT\rangle_n^= \sqsubseteq \langle FT\rangle_n^= \circ_1 \langle \mathbf{I}_T\rangle_F$$

by Substitution Lemma (Proposition 4.13). Lastly, by using $\mathbf{Par}(FT)$, for each x of type FT, we have $\langle FT\rangle_n^= x \ldots x$, thus we have $\langle T\rangle_n^=(\langle \mathbf{I}_T\rangle_F x)x \ldots x$, which implies $x = \langle \mathbf{I}_T\rangle_F x$. $\square$

4.3. Fixed Point Type

In this subsection, we define some notations for fixed point types.

Definition 4.17. For a functor F, $\mu F :\equiv \mu\gamma.F\gamma$, $\nu F :\equiv \nu\gamma.F\gamma$.

Definition 4.18. For a functor F and a type T,

$$\begin{array}{ll}
\boldsymbol{c}_{\mu F,T} :\equiv \lambda xy.yTx & \text{of type } (FT \to T) \to \mu F \to T, \\
\boldsymbol{f}_{\mu F} :\equiv \lambda x\Lambda\gamma.\lambda y.y(\langle \boldsymbol{c}_{\mu F,\gamma} y\rangle_F x) & \text{of type } F\mu F \to \mu F, \\
\boldsymbol{g}_{\mu F} :\equiv \boldsymbol{c}_{\mu F, F\mu F}\langle \boldsymbol{f}_{\mu F}\rangle_F & \text{of type } \mu F \to F\mu F, \\
\boldsymbol{c}_{\nu F,T} :\equiv \lambda xy.\Lambda\gamma.\lambda z.zTxy & \text{of type } (T \to FT) \to T \to \nu F, \\
\boldsymbol{g}_{\nu F} :\equiv \lambda x.x(F\nu F)\Lambda\gamma\lambda yz.\langle \boldsymbol{c}_{F\nu F,\gamma} y\rangle_F (yz) & \text{of type } \nu F \to F\nu F, \\
\boldsymbol{f}_{\nu F} :\equiv \boldsymbol{c}_{\nu F, F\nu F}\langle \boldsymbol{g}_{\nu F}\rangle_F & \text{of type } F\nu F \to \nu F.
\end{array}$$

We often omit the type suffixes, that is, $\boldsymbol{c}$ and $\boldsymbol{c}_T$ are sometimes written for $\boldsymbol{c}_{\mu F,T}$, and $\boldsymbol{f}$ is written for $\boldsymbol{f}_{\mu F}$, and so forth.

The terms $\boldsymbol{c}_{\mu F,T}$ and $\boldsymbol{c}_{\nu F,T}$ are the standard encodings of the destructor of μF and the constructor of νF. Also $\boldsymbol{f}_{\mu F}$ and $\boldsymbol{f}_{\nu F}$ are the standard encodings of the fold operators. Similarly, $\boldsymbol{g}_{\mu F}$ and $\boldsymbol{g}_{\nu F}$ are the standard encodings of the unfold operators.

We have those equations such that $\boldsymbol{c}_{\mu F,T} X \circ \boldsymbol{f}_{\mu F} =_\beta X \circ \langle \boldsymbol{c}_{\mu F} X\rangle_F$ for each term $X^{FT\to T}$, and also $\boldsymbol{g}_{\nu F} \circ \boldsymbol{c}_{\mu F,T} X =_\beta \langle \boldsymbol{c}_{\mu F} X\rangle_F \circ X$ for each term $X^{T \to FT}$. These are standard encodings of calculations of recursive structures and lazy structures. It is well-known that the axioms of parametricities prove that μF is a initial algebra and νF is a final coalgebra [6, 7]. We will explain that in our formal system $\mathbf{L}$.

Proposition 4.19. (Fundamental Conformities) Let Θ be an assignment such that $\Theta \equiv \{\Phi_1/\gamma_1, \ldots, \Phi_m/\gamma_m\}$ where Φ_j is a predicate of predicate type $T_{1,j} \to \cdots \to T_{n,j} \to *$ for $1 \le j \le m$. Let θ_i be a substitution of types such that $\theta_i \equiv [T_{i,1}/\gamma_1, \ldots, T_{i,m}/\gamma_m]$ for $1 \le i \le n$. Let F be a functor and $F_i \equiv F\theta_i$ for each i such that $1 \le i \le n$. Then $\mathbf{L}$ derives the following:

1. $Env(\Theta) \vdash \langle F\mu F \to \mu F\rangle_n\{\Theta\}\boldsymbol{f}_{\mu F_1}\cdots\boldsymbol{f}_{\mu F_n}$
2. $Env(\Theta) \vdash \langle \nu F \to F\nu F\rangle_n\{\Theta\}\boldsymbol{g}_{\nu F_1}\cdots\boldsymbol{g}_{\nu F_n}$
3. $Env(\Theta) \vdash \langle (FT \to T) \to \mu F \to T\rangle_n\{\Theta\}\boldsymbol{c}_{\mu F_1, T\theta_1}\cdots\boldsymbol{c}_{\mu F_n, T\theta_n}$
4. $Env(\Theta) \vdash \langle (T \to FT) \to T \to \nu F\rangle_n\{\Theta\}\boldsymbol{c}_{\nu F_1, T\theta_1}\cdots\boldsymbol{c}_{\nu F_n, T\theta_n}$
5. $Env(\Theta) \vdash \langle \mu F \to \mu F\rangle_n\{\Theta\}(\boldsymbol{f}_{\mu F_1}\circ\boldsymbol{g}_{\mu F_1})\mathbf{I}_{\mu F_2}\ldots\mathbf{I}_{\mu F_n}$
6. $Env(\Theta) \vdash \langle \nu F \to \nu F\rangle_n\{\Theta\}(\boldsymbol{f}_{\nu F_1}\circ\boldsymbol{g}_{\nu F_1})\mathbf{I}_{\nu F_2}\ldots\mathbf{I}_{\nu F_n}$
7. $Env(\Theta) \vdash \langle F\mu F \to F\mu F\rangle_n\{\Theta\}(\boldsymbol{g}_{\mu F_1}\circ\boldsymbol{f}_{\mu F_1})\mathbf{I}_{F_2\mu F_2}\ldots\mathbf{I}_{F_n\mu F_n}$
8. $Env(\Theta) \vdash \langle F\nu F \to F\nu F\rangle_n\{\Theta\}(\boldsymbol{g}_{\nu F_1}\circ\boldsymbol{f}_{\nu F_1})\mathbf{I}_{F_2\nu F_2}\ldots\mathbf{I}_{F_n\nu F_n}$
9. $Env(\Theta) \vdash \langle \mu F \to \mu F\rangle_n\{\Theta\}(\boldsymbol{c}_{\mu F_1}\boldsymbol{f}_{\mu F_1})\mathbf{I}_{\mu F_2}\ldots\mathbf{I}_{\mu F_n}$
10. $Env(\Theta) \vdash \langle \nu F \to \nu F\rangle_n\{\Theta\}(\boldsymbol{c}_{\nu F_1}\boldsymbol{g}_{\nu F_1}\circ\boldsymbol{f}_{\nu F_1}\circ\boldsymbol{g}_{\nu F_1})\mathbf{I}_{\nu F_2}\ldots\mathbf{I}_{\nu F_n}$

Proof:
All the proofs are done by straightforward calculations with the Functor Lemma (Proposition 4.15). We will only show the proof of 6. $\langle \nu F \to \nu F\rangle_n\{\Theta\}(\boldsymbol{f}_{\nu F_1}\circ\boldsymbol{g}_{\nu F_1})\mathbf{I}_{\nu F_2}\ldots\mathbf{I}_{\nu F_n}$. Others are similarly shown.
Step 0. Put $h_F :\equiv \Lambda\gamma.\lambda x^{\gamma\to F\gamma}y^\gamma.\langle \boldsymbol{c}_{\nu F,\gamma}x\rangle_F(xy)$ of type $\Pi\gamma.(\gamma \to F\gamma) \to \gamma \to F\nu F$. Then, we have $\boldsymbol{g}x =_\beta x(F\nu F)h_F$ and $\boldsymbol{g}(\boldsymbol{c}_T xy) =_\beta h_F Txy$.
Step 1. Let $\gamma_1, \ldots, \gamma_n$ be type variables and ϕ be a predicate variable of predicate type $\gamma_1 \to \cdots \to \gamma_n \to *$. Let $v_1, \ldots, v_n, w_1, \ldots, w_n$ be variables where v_i is of type $\gamma_i \to F_i\gamma_i$ and w_i is of type γ_i for $1 \le i \le n$. Put $\Phi :\equiv \lambda uw_2\ldots w_n.\exists w_1.u = \boldsymbol{g}_{\nu F_1}(\boldsymbol{c}_{\gamma_1}v_1w_1) \wedge \phi w_1..w_n$. Suppose

$$\langle \gamma \to F\gamma\rangle_n\{\phi/\gamma, \Theta\}v_1\ldots v_n \text{ and } \phi w_1\ldots w_n.$$

Then we will prove that

$$\langle\gamma \to F\gamma\rangle_n\{\Phi/\gamma,\Theta\}\langle\boldsymbol{g}\rangle_{F_1}v_2\ldots v_n \text{ and } \Phi(\boldsymbol{g}(\boldsymbol{c}v_1w_1))w_2\ldots w_n.$$

First, we immediately have $\Phi(\boldsymbol{g}(\boldsymbol{c}v_1w_1))\ldots w_n$ by the definition of Φ. Next, we have $\phi \sqsubseteq \Phi \circ_1 (\boldsymbol{g}\circ \boldsymbol{c}v_1)$. Thus, by the Functor Lemma (Proposition 4.15), we have $F^\Theta\phi \sqsubseteq F^\Theta\Phi \circ_1 \langle \boldsymbol{g}\circ\boldsymbol{c}v_1\rangle_{F_1}$.

Now, suppose $\Phi u_1\ldots u_n$, then there is w such that $u_1 = \boldsymbol{g}(\boldsymbol{c}v_1w)$ and $\phi w u_2\ldots u_n$. Thus

$$F^\Theta\phi(v_1w)(v_2u_2)\ldots(v_nu_n).$$

Hence

$$F^\Theta\Phi(\langle\boldsymbol{g}\circ\boldsymbol{c}v_1\rangle_{F_1}(v_1w))(v_2u_2)\ldots(v_nu_n).$$

On the other hand, $\langle\boldsymbol{g}\circ\boldsymbol{c}v_1\rangle(v_1w) =_\beta (\langle\boldsymbol{g}\rangle\circ\langle\boldsymbol{c}v_1\rangle)(v_1w) =_\beta \langle\boldsymbol{g}\rangle(\langle\boldsymbol{c}v_1\rangle(v_1w)) =_\beta \langle\boldsymbol{g}\rangle(\boldsymbol{g}(\boldsymbol{c}v_1w))$ $= \langle\boldsymbol{g}\rangle u_1$. Hence we have

$$F^\Theta\Phi(\langle\boldsymbol{g}\rangle u_1)(v_2u_2)\ldots(v_nu_n).$$

Therefore we have

$$\langle\gamma\to F\gamma\rangle_n\{\Phi/\gamma,\Theta\}\langle\boldsymbol{g}\rangle v_2\ldots v_n.$$

Step 2. Let $\delta_1,\ldots,\delta_n$ be type variables and ψ be a predicate variable of predicate type $\delta_1\to\cdots\to\delta_n\to *$. Let $y_1,\ldots,y_n$ be variables where y_i is of type $(\gamma_i\to F_i\gamma_i)\to\gamma_i\to\delta_i$. Put $\Psi :\equiv \psi\circ_1 y_1\nu F_1\langle\boldsymbol{g}_{F_1}\rangle_{F_1}$. Suppose that

$$\langle\Pi\gamma.(\gamma\to F\gamma)\to\gamma\to\delta\rangle_n\{\psi/\delta,\Theta\}y_1\ldots y_n,$$

then we will prove that

$$\langle\Pi\gamma.(\gamma\to F\gamma)\to\gamma\to\delta\rangle_n\{\Psi/\delta,\Theta\}h_{F_1}y_2\ldots y_n.$$

Assume

$$\langle\gamma\to F\gamma\rangle_n\{\phi/\gamma,\Theta\}v_1\ldots v_n \text{ and } \phi w_1\ldots w_n.$$

Then, by Step 1, we have

$$\langle\gamma\to F\gamma\rangle_n\{\Phi/\gamma,\Theta\}\langle\boldsymbol{g}\rangle v_2\ldots v_n \text{ and } \Phi(h\gamma_1v_1w_1)\ldots w_n.$$

By the supposition, we have

$$\langle(\gamma\to F\gamma)\to\gamma\to\delta\rangle_n\{\Phi/\gamma,\psi/\delta,\Theta\}(y_1\nu F_1)\ldots(y_n\gamma_n).$$

Thus

$$\psi(y_1\nu F_1\langle\boldsymbol{g}\rangle(\boldsymbol{g}(\boldsymbol{c}v_1w_1)))(y_2\gamma_2v_2w_2)\ldots(y_n\gamma_nv_nw_n).$$

Thus

$$\Psi(h_{F_1}\gamma_1v_1w_1)(y_2\gamma_2v_2w_2)\ldots(y_n\gamma_nv_nw_n),$$

because $\boldsymbol{g}(\boldsymbol{c}v_1w_1) =_\beta h_{F_1}\gamma_1v_1w_1$. Hence we have

$$\langle(\gamma\to F\gamma)\to\gamma\to\delta\rangle_n\{\phi/\gamma,\Psi/\delta,\Theta\}(h_{F_1}\gamma_1)(y_2\gamma_2)\ldots(y_n\gamma_n).$$

Therefore we have

$$\langle\Pi\gamma.(\gamma\to F\gamma)\to\gamma\to\delta\rangle_n\{\Psi/\delta,\Theta\}h_{F_1}y_2\ldots y_n.$$

Step 3. Suppose that

$$\langle\nu F\rangle_n\{\Theta\}x_1\ldots x_n.$$

We will show that

$$\langle\nu F\rangle_n\{\Theta\}(\boldsymbol{f}(\boldsymbol{g}x_1))x_2\ldots x_n,$$

which is identical to

$$\langle \nu F\rangle_n\{\Theta\}(\boldsymbol{f}(x_1(F_1\nu F_1)h_{F_1}))x_2\dots x_n.$$

Assume that

$$\langle \Pi\gamma.(\gamma\to F\gamma)\to\gamma\to\delta\rangle_n\{\psi/\delta,\Theta\}y_1\dots y_n,$$

then our target is $\psi(\boldsymbol{f}(x_1(F_1\nu F_1)h_{F_1})\delta_1 y_1)(x_2\delta_2 y_2)\dots(x_n\delta_n y_n)$.

The supposition is expanded into

$$\forall\delta_1\dots\delta_n.\forall\psi.\langle(\Pi\gamma.(\gamma\to T\gamma)\to\gamma\to\delta)\to\delta\rangle_n\{\psi/\delta,\Theta\}(x_1\delta_1)\dots(x_n\delta_n).$$

Thus we have

$$\langle(\Pi\gamma.(\gamma\to T\gamma)\to\gamma\to\delta)\to\delta\rangle_n\{\Psi/\delta,\Theta\}(x_1\delta_1)\dots(x_n\delta_n).$$

By the assumption and Step 2, we have

$$\langle \Pi\gamma.(\gamma\to T\gamma)\to\gamma\to\delta\rangle_n\{\Psi/\delta,\Theta\}h_{F_1}y_2\dots y_n.$$

Thus we have

$$\Psi(x_1(F_1\nu F_1)h_{F_1})(x_2\delta_2 y_2)\dots(x_n\delta_n y_n),$$

which is expanded into

$$\psi(y_1\nu F_1\langle\boldsymbol{g}\rangle_{F_1}(x_1(F_1\nu F_1)h_{F_1}))(x_2\delta_2 y_2)\dots(x_n\delta_n y_n).$$

We also have $y_1\nu F_1\langle\boldsymbol{g}\rangle_{F_1}(x_1(F_1\nu F_1)h_{F_1}) =_\beta \boldsymbol{f}(x_1(F_1\nu F_1)h_{F_1})\delta_1 y_1$. Thus we obtain the goal:

$$\psi(\boldsymbol{f}(x_1(F_1\nu F_1)h_{F_1})\delta_1 y_1)(x_2\delta_2 y_2)\dots(x_n\delta_n y_n).$$

This completes the proof. □

Remember that the notation $\simeq$ is defined in Definition 2.4 as: $X^{T\to U}\simeq Y^{T\to U} :\equiv \forall z^T.Xz = Yz$.

Definition 4.20. Let T be μF or νF, then $\mathbf{Iso}(T) :\equiv \mathbf{I}_T \simeq \boldsymbol{f}\circ\boldsymbol{g}$, $\mathbf{Iso}'(T) :\equiv \mathbf{I}_F T\simeq \boldsymbol{g}\circ\boldsymbol{f}$.

$\mathbf{Iso}(T)$ and $\mathbf{Iso}'(T)$ say that $\boldsymbol{f}$ and $\boldsymbol{g}$ are isomorphisms between T and FT.

Corollary 4.21. (Lambek Lemma) If $T=\mu F$ or $T=\nu F$, then $\mathbf{Par}_n(T)\vdash\mathbf{Iso}(T)$, and $\mathbf{Par}_n(FT)\vdash\mathbf{Iso}'(T)$.

Proof:
By $\mathbf{Par}_n(T)$, we have $\langle T\rangle_n^{=}x\dots x$. Then we have $\langle T\rangle_n^{=}((\boldsymbol{f}\circ\boldsymbol{g})x)\dots x$ by the fundamental conformity (Proposition 4.19). Therefore, we have $x=(\boldsymbol{f}\circ\boldsymbol{g})x$ by $\mathbf{Par}_n(T)$. The formula $\mathbf{Iso}'$ is similarly proved. □

Definition 4.22. $\mathbf{Id}(\mu F) :\equiv \mathbf{I}_{\mu F}\simeq \boldsymbol{cf}$, $\mathbf{Id}(\nu F) :\equiv \mathbf{I}_{\mu F}\simeq \boldsymbol{cg}$,

Corollary 4.23. If $T=\mu F$ or $T=\nu F$, then $\mathbf{Par}_n(T)\vdash\mathbf{Id}(T)$.

Proof:
Similar to Corollary 4.21. □

4.4. Fixed Point Predicate

In this subsection, we define fixed point predicates, and give some of their properties. Finally we gave the fixed point lemma (Corollary 4.30). Remember that $\sqsubseteq$ and $\cong$ are defined in Definition 2.4 as

$$\Phi \sqsubseteq \Psi :\equiv \forall x_1 \dots x_n.\Phi x_1 \dots x_n \supset \Psi x_1 \dots x_n, \quad \text{and} \quad \Phi \cong \Psi :\equiv \Phi \sqsubseteq \Psi \wedge \Psi \sqsubseteq \Phi.$$

Definition 4.24. Let $\Phi[\phi^A]$ be a predicate of predicate type $A \equiv T_1 \to \cdots T_n \to *$, where ϕ occurs only positively. Then:

$$\mu\phi.\Phi[\phi] :\equiv \lambda x_1 \dots x_n.\forall\phi.(\Phi[\phi] \sqsubseteq \phi) \supset \phi x_1 \dots x_n,$$

$$\nu\phi.\Phi[\phi] :\equiv \lambda x_1 \dots x_n.\exists\phi.(\phi \sqsubseteq \Phi[\phi]) \wedge \phi x_1 \dots x_n.$$

We call $\mu\phi.\Phi[\phi]$ the *least fixed point predicate*, and $\nu\phi.\Phi[\phi]$ the *greatest fixed point predicate*.

Proposition 4.25. (Inferences of Fixed Point Predicate)

$$\mathit{Env}(\Phi,\Psi), \Phi[\Psi] \sqsubseteq \Psi \vdash \mu\phi.\Phi[\phi] \sqsubseteq \Psi, \quad \mathit{Env}(\Phi) \vdash \mu\phi.\Phi[\phi] \cong \Phi[\mu\phi.\Phi[\phi]],$$

$$\mathit{Env}(\Phi,\Psi), \Psi \sqsubseteq \Phi[\Psi] \vdash \Psi \sqsubseteq \nu\phi.\Phi[\phi], \quad \mathit{Env}(\Phi) \vdash \nu\phi.\Phi[\phi] \cong \Phi[\nu\phi.\Phi[\phi]].$$

We define some notations before we prove the fixed point lemma (Corollary 4.30). First we define a notation $T^n \to *$. It denotes a predicate type which has n occurrences of $T \to$ before $*$. Namely, $T^2 \to * :\equiv T \to T \to *$, and $T^3 \to * :\equiv T \to T \to T \to *$. The formal definition is as below.

Definition 4.26. $T^0 \to * :\equiv *, \quad T^{n+1} \to * :\equiv T \to T^n \to *.$

Definition 4.27. Let Φ be a predicate of predicate type $T^n \to *$, and X be a term of type $U \to T$. Then, $\Phi \circ X :\equiv \lambda x_1^U \dots x_n^U.\Phi(Xx_1)\dots(Xx_n)$.

Definition 4.28. Let F be a functor, $\phi_1, \dots, \phi_m$ be predicate variables where ϕ_i is of predicate type $T_i^n \to *$, let θ be a substitution of types $[T_1/\gamma_1, \dots, T_m/\gamma_m]$, and Θ be a assignment $\{\phi_1/\gamma_1, \dots, \phi_m/\gamma_m\}$. Then

$$\mathbf{Fix}_n(\mu F, \theta) :\equiv \forall\phi_1 \dots \phi_m.(\langle\mu F\rangle_n\{\Theta\} \cong \mu\psi^{\mu F\theta^n \to *}.(F^\Theta\psi) \circ \boldsymbol{g}_{\mu F\theta})$$

$$\mathbf{Fix}_n(\nu F, \theta) :\equiv \forall\phi_1 \dots \phi_m.(\langle\nu F\rangle_n\{\Theta\} \cong \mu\psi^{\nu F\theta^n \to *}.(F^\Theta\psi) \circ \boldsymbol{g}_{\nu F\theta})$$

We call $\mathbf{Fix}_n(T,\theta)$ *fixed point property* of T for n-ary predicates.

The intuitive meaning of fixed point property is that the conformity of fixed point type is equivalent to the fixed point of conformity.

Lemma 4.29. **L** derives the following:

$$\mathbf{Id}(\mu F\theta), \mathbf{Iso}(\mu F\theta), \mathbf{Iso}'(\mu F\theta) \vdash \mathbf{Fix}_n(\mu F, \theta),$$

$$\mathbf{Id}(\mu F\theta), \mathbf{Iso}(\mu F\theta), \mathbf{Iso}'(\mu F\theta) \vdash \mathbf{Fix}_n(\nu F, \theta).$$

Proof:

We only show the assertion for μ. The proof for ν is extremely equivalent.

First we will show $\langle\mu F\rangle_n\{\Theta\} \sqsubseteq \mu\phi.F^\Theta\phi \circ \boldsymbol{g}$.

By the inference of fixed point predicate (Proposition 4.25),

$$F^\Theta(\mu\phi.F^\Theta\phi \circ \boldsymbol{g}) \circ \boldsymbol{g} \sqsubseteq \mu\phi.F^\Theta\phi \circ \boldsymbol{g}.$$

By $\mathbf{Iso}'(\mu F\theta)$,

$$F^{\Theta}(\mu\phi.F^{\Theta}\phi \circ \boldsymbol{g}) \sqsubseteq (\mu\phi.F^{\Theta}\phi \circ \boldsymbol{g}) \circ \boldsymbol{f},$$

which is equivalent to

$$\langle F\gamma \to \gamma\rangle_n\{(\mu\phi.F^{\Theta}\phi \circ \boldsymbol{g})/\gamma, \Theta\}\boldsymbol{f} \dots \boldsymbol{f}.$$

By 1. and 2. of the fundamental conformity (Proposition 4.19),

$$\langle \mu F \to \gamma\rangle_n\{(\mu\phi.F^{\Theta}\phi \circ \boldsymbol{g})/\gamma, \Theta\}(\boldsymbol{cf}) \dots (\boldsymbol{cf}).$$

which is equivalent to

$$\langle \mu F\rangle_n\{\Theta\} \sqsubseteq (\mu\phi.F^{\Theta}\phi \circ \boldsymbol{g}) \circ \boldsymbol{cf}.$$

By $\mathbf{Id}(\mu F\theta)$, we have the goal: $\langle \mu F\rangle_n\{\Theta\} \sqsubseteq \mu\phi.F^{\Theta}\phi \circ \boldsymbol{g}$.

Next we will show $\mu\phi.F^{\Theta}\phi \circ \boldsymbol{g} \sqsubseteq \langle \mu F\rangle_n\{\Theta\}$.

By Substitution Lemma (Proposition 4.13) and 1. of the fundamental conformities (Proposition 4.19),

$$F^{\Theta}(\langle \mu F\rangle_n\{\Theta\}) \sqsubseteq \langle \mu F\rangle_n\{\Theta\} \circ \boldsymbol{f}.$$

By $\mathbf{Iso}(\mu F\theta)$,

$$F^{\Theta}(\langle \mu F\rangle_n\{\Theta\}) \circ \boldsymbol{g} \sqsubseteq \langle \mu F\rangle_n\{\Theta\}.$$

By the inference of fixed point predicate (Proposition 4.25), we have the goal: $\mu\phi.F^{\Theta}\phi \circ \boldsymbol{g} \sqsubseteq \langle \mu F\rangle_n\{\Theta\}$. □

Corollary 4.30. (Fixed Point Lemma) Let F be a functor in R, and $F\theta$ also be a functor in R. Then $\mathbf{Par}_m(\mathsf{R}) \vdash \mathbf{Fix}_n(\mu F, \theta)$ and $\mathbf{Par}_m(\mathsf{R}) \vdash \mathbf{Fix}_n(\nu F, \theta)$ for any $m, n \geq 2$.

Proof:

By Lemmata 4.21, 4.23, and 4.29. □

Note that we can choose the suffix numbers m and n independently. In [8], this lemma is mentioned, but it is restricted to the same $m = n$.

4.5. Crucial Lemma

This subsection is spent on proving Lemma 4.37. It is crucial for the proof of our main theorem 4.6.

Definition 4.31. Let Φ be a predicate of predicate type $T^n \to *$ for $n \geq 2$. Then:

$$\hat{\mathbf{b}}[\Phi] \equiv \lambda x^T y^T.(\Phi x \dots x \wedge \bigwedge_{i=1,\dots,n} \forall z_1 \dots z_{i-1} z_{i+1} \dots z_n. \\ \Phi z_1 \dots z_{i-1} x z_{i+1} \dots z_n \supset\subset \Phi z_1 \dots z_{i-1} y z_{i+1} \dots z_n),$$

$$\bar{\mathbf{b}}[\Phi] :\equiv \lambda x^T y^T.\exists z_1 \dots z_n.(\Phi z_1 \dots z_n \wedge (\bigvee_{i=1,\dots,n} x = z_i) \wedge (\bigvee_{i=1,\dots,n} y = z_i)),$$

$$\check{\mathbf{b}}[\Phi] :\equiv \mu\phi^{T\to T\to *}.\lambda xy.\exists z.\bar{\mathbf{b}}[\Phi]xz \wedge (z = y \vee \phi zy).$$

The binary relation $\hat{\mathbf{b}}[\Phi]$ denotes a partial equivalence relation in a domain in which Φ holds. If $\hat{\mathbf{b}}[\Phi]xy$, then x and y behave in the same way for Φ. The binary relation $\bar{\mathbf{b}}[\Phi]$ is the symmetric closure of Φ, and $\check{\mathbf{b}}[\Phi]$ is the transitive closure of $\bar{\mathbf{b}}[\Phi]$.

Definition 4.32. Let m and n be number such as ≥ 2, and Φ be a predicate of predicate type $T^n \to *$. Then:

$$\mathbf{C}_n^m[\Phi] :\equiv \lambda x_1 \dots x_m.\hat{\mathbf{b}}[\Phi]x_1x_2 \wedge \hat{\mathbf{b}}[\Phi]x_1x_3 \wedge \dots \wedge \hat{\mathbf{b}}[\Phi]x_1x_m,$$

$$\mathbf{D}_n^m[\Phi] :\equiv \lambda x_1 \dots x_m.\check{\mathbf{b}}[\Phi]x_1x_2 \wedge \check{\mathbf{b}}[\Phi]x_1x_3 \wedge \dots \wedge \check{\mathbf{b}}[\Phi]x_1x_m.$$

$\mathbf{C}_n^m[\ ]$ and $\mathbf{D}_n^m[\ ]$ are partial equivalence relations of arity m. The denotation of $\mathbf{C}_n^m[\Phi]$ is an internal relation over the domain of Φ, and $\mathbf{D}_n^m[\Phi]$ is the closure of Φ in a generalized sense. If Φ already denotes a partial equivalence relation, $\mathbf{C}_n^n[\ ]$ and $\mathbf{D}_n^n[\ ]$ do not move Φ. Thus, the next proposition holds.

Proposition 4.33.

$$\mathit{Env}(\Phi) \vdash \mathbf{C}_n^n[\Phi] \sqsubseteq \Phi, \quad \mathit{Env}(\Phi) \vdash \mathbf{D}_n^l[\Phi] \cong \mathbf{C}_m^l[\mathbf{D}_n^m[\Phi]],$$

$$\mathit{Env}(\Phi) \vdash \Phi \sqsubseteq \mathbf{D}_n^n[\Phi], \quad \mathit{Env}(\Phi) \vdash \mathbf{D}_n^l[\Phi] \cong \mathbf{C}_m^l[\mathbf{D}_n^m[\Phi]].$$

Moreover, we have the following property.

Proposition 4.34. $\mathit{Env}(\Phi) \vdash (\Phi \sqsubseteq \mathbf{C}_n^n[\Phi]) \supset\subset (\mathbf{D}_n^n[\Phi] \sqsubseteq \Phi)$

Proof:
Easy. $\square$

Proposition 4.35. $\mathbf{D}[\ ]$ is a monotone context, that is, ϕ occurs only positively in $\mathbf{D}_n^m[\phi]$. Thus $\mathbf{L}$ derives:

$$\mathit{Env}(\Phi, \Psi), \Phi \sqsubseteq \Psi \vdash \mathbf{D}[\Phi] \sqsubseteq \mathbf{D}[\Psi].$$

Proof:
Obvious. $\square$

Definition 4.36. Let Θ be an assignment $\{\Phi_1^{T_1^n \to *}/\gamma_1, \dots, \Phi_l^{T_l^n \to *}/\gamma_l\}$. Then:

$$\mathbf{C}_n^m[\Theta] :\equiv \{\mathbf{C}_n^m[\Phi_1]/\gamma_1, \dots, \mathbf{C}_n^m[\Phi_l]/\gamma_l\},$$

$$\mathbf{D}_n^m[\Theta] :\equiv \{\mathbf{D}_n^m[\Phi_1]/\gamma_1, \dots, \mathbf{D}_n^m[\Phi_l]/\gamma_l\}.$$

Lemma 4.37. (Crucial Lemma) Let $T, T_1, \dots, T_i, T'_1, \dots, T'_{i'}, U_1, \dots, U_j, U'_1, \dots, U'_{j'}$ be types in R. Suppose that type variables $\gamma_1, \dots, \gamma_i, \gamma'_1, \dots, \gamma'_{i'}$ occur only positively in T, and $\delta_1, \dots, \delta_j, \delta'_1, \dots, \delta'_{j'}$ occur only negatively in T. Let

$$\Phi_1{}^{T_1^m \to *}, \dots, \Phi_i{}^{T_i^m \to *}, \Phi'_1{}^{T'^n_1 \to *}, \dots, \Phi'_{i'}{}^{T'^n_{i'} \to *},$$

$$\Psi_1{}^{U_1^n \to *}, \dots, \Psi_j{}^{U_j^n \to *}, \Psi'_1{}^{U'^m_1 \to *}, \dots, \Psi'_{j'}{}^{U'^m_{j'} \to *}$$

be predicates. Let assignments Θ, Θ', Ξ and Ξ' be defined as:

$$\Theta :\equiv \{\Phi_1/\gamma_1, \dots, \Phi_i/\gamma_i\}, \quad \Theta' :\equiv \{\Phi'_1/\gamma'_1, \dots, \Phi'_{i'}/\gamma'_{i'}\},$$

$$\Xi :\equiv \{\Psi_1/\delta_1, \dots, \Psi_j/\delta_j\}, \ \ \Xi' :\equiv \{\Psi'_1/\delta'_1, \dots, \Psi'_{j'}/\delta'_{j'}\}.$$

Then for arbitrary numbers $k, l, m, n \geq 2$, it holds that:

$$\begin{aligned}&\mathit{Env}(\Theta, \Theta', \Xi, \Xi'), \mathbf{Par}_l(\mathsf{R}) \vdash \\ &\qquad \mathbf{D}_m^k[\langle T \rangle_m\{\Theta, \mathbf{C}_n^m[\Theta'], \mathbf{D}_n^m[\Xi], \Xi'\}] \sqsubseteq \mathbf{C}_n^k[\langle T \rangle_n\{\mathbf{D}_m^n[\Theta], \Theta', \Xi, \mathbf{C}_m^n[\Xi']\}].\end{aligned}$$

Proof:
The proof is by induction on the construction of T. We use the following notations:

$$M :\equiv \{\Theta, \mathbf{C}^m_n[\Theta'], \mathbf{D}^m_n[\Xi], \Xi'\}, \quad \text{and,} \quad N :\equiv \{\mathbf{D}^n_m[\Theta], \Theta', \Xi, \mathbf{C}^n_m[\Xi']\}.$$

M is an assignment of predicates with m arguments, and N is an assignment of predicates with n arguments. The substitution θ is defined as:

$$\theta :\equiv [T_1/\gamma_1, \ldots, T_i/\gamma_i, T'_1/\gamma'_1, \ldots, T'_{i'}/\gamma_{i'}, U_1/\delta_1, \ldots, U_j/\delta_j, U'_1/\delta'_1, \ldots, U'_{j'}/\delta_{j'}]$$

For an arbitrary type T, $\langle T\rangle_m\{M\}$ is a predicate of predicate type $T\theta^m \to *$, and $\langle T\rangle_n\{N\}$ is a predicate of predicate type $T\theta^n \to *$.

We show the proof step only for fixed point types and for functional types. The steps for unit type, pair type, and sum type are easy.

– The step for least fixed point types μF is shown by using the fixed point lemma (Corollary 4.30) as follows.

By the fixed point lemma, we have

$$\langle \mu F\rangle_m\{M\} \cong \mu\phi.F^M\phi \circ \boldsymbol{g}_{\mu F\theta}, \quad \text{and} \quad \langle \mu F\rangle_n\{N\} \cong \mu\phi.F^N\phi \circ \boldsymbol{g}_{\mu F\theta}.$$

The induction hypothesis is $F^M(\mathbf{C}^m_n[\langle \mu F\rangle_n\{N\}]) \sqsubseteq \mathbf{C}^m_n[F^N(\langle \mu F\rangle_n\{N\})]$, thus

$$F^M(\mathbf{C}^m_n[\langle \mu F\rangle_n\{N\}]) \circ \boldsymbol{g}_{\mu F\theta} \sqsubseteq \mathbf{C}^m_n[\langle \mu F\rangle_n\{N\}].$$

Therefore we have $\langle \mu F\rangle_m\{M\} \sqsubseteq \mathbf{C}^n_n[\langle \mu F\rangle_n\{N\}]$, by the inference of fixed point predicate (Proposition 4.25). By operating $\mathbf{D}^k_m[\,]$ on both side, we obtain the goal:

$$\mathbf{D}^k_m[\langle \mu F\rangle_m\{M\}] \sqsubseteq \mathbf{C}^k_n[\langle \mu F\rangle_n\{N\}].$$

– The proof of the step for the greatest fixed point type νF is as below.

Just similar to the step for the least fixed point type, we have

$$\mathbf{D}^n_m[\langle \nu F\rangle_m\{M\}] \sqsubseteq \langle \nu F\rangle_n\{N\}$$

In contrast to $\mathbf{D}[\,]$, the context $\mathbf{C}[\,]$ is not monotonic. So we cannot get $\mathbf{D}^k_m[\langle \nu F\rangle_m\{M\}] \sqsubseteq \mathbf{C}^k_n[\langle \nu F\rangle_n\{N\}]$ directly. Hence we take one more step. As the special case for $m = n$, we have

$$\mathbf{D}^n_n[\langle \nu F\rangle_n\{N\}] \sqsubseteq \langle \nu F\rangle_n\{N\}.$$

Thus, by Proposition 4.34, we have $\langle \nu F\rangle_n\{N\} \sqsubseteq \mathbf{C}^n_n[\langle \nu F\rangle_n\{N\}]$. Therefore we obtain

$$\mathbf{D}^n_m[\langle \nu F\rangle_m\{M\} \sqsubseteq \mathbf{C}^n_n[\langle \nu F\rangle_n\{N\}].$$

– The proof of the step for a functional type $T \to U$ is as below.

Our goal in this step is to derive

$$\mathbf{D}^k_m[\langle T \to U\rangle_m\{M\}] \sqsubseteq \mathbf{C}^k_n[\langle T \to U\rangle_n\{N\}].$$

It is enough to show $\bar{\mathbf{b}}[\langle T \to U\rangle_m\{M\}] \sqsubseteq \hat{\mathbf{b}}[\langle T \to U\rangle_n\{N\}]$, because then we get the goal formula by operating with $\mathbf{D}^k_2[\,]$ on both sides. The induction hypotheses are:

$$\mathbf{D}^k_n[\langle T\rangle_n\{N\}] \sqsubseteq \mathbf{C}^k_m[\langle T\rangle_m\{M\}], \quad \mathbf{D}^k_m[\langle U\rangle_m\{M\}] \sqsubseteq \mathbf{C}^k_n[\langle U\rangle_n\{N\}].$$

Under the assumption $\bar{\mathbf{b}}[\langle T \to U\rangle_m\{M\}]xy$, first we will show how to obtain

$$\langle T \to U\rangle_n\{N\}x \ldots x,$$

and next we will show how to obtain

$$\forall z_1 \dots z_{i-1} z_{i+1} \dots z_n . \langle T \to U \rangle_n \{N\} z_1 \dots z_{i-1} x z_{i+1} \dots z_n \supset\subset \langle T \to U \rangle_n \{N\} z_1 \dots z_{i-1} y z_{i+1} \dots z_n$$

for each i.

Now we will show $\langle T \to U \rangle_n \{N\} x \dots x$, that is, we will derive $\langle U \rangle_n \{N\} (x w_1) \dots (x w_n)$ from $\langle T \rangle_n \{N\} w_1 \dots w_n$. Suppose that $\langle T \rangle_n \{N\} w_1 \dots w_n$, then, $\mathbf{D}^m_n[\langle T \rangle_n \{N\}] w_{\sigma 1} \dots w_{\sigma m}$ for each mapping σ of $\{1, \dots, m\}$ into $\{1, \dots, n\}$.

By the induction hypothesis, we have

$$\mathbf{C}^m_m[\langle T \rangle_m \{M\}] w_{\sigma 1} \dots w_{\sigma m}, \quad \text{thus} \quad \langle T \rangle_m \{M\} w_{\sigma 1} \dots w_{\sigma m}.$$

From the assumption, there are a number $i \in \{1, \dots, m\}$ and a sequence of $z_1, z_2, \dots$ such that $x = z_i \wedge \langle T \to U \rangle_m \{M\} z_1 \dots z_m$. Thus

$$x w_{\sigma i} = z_i w_{\sigma i} \wedge \langle U \rangle_m \{M\} (z_1 w_{\sigma 1}) \dots (z_m w_{\sigma m}).$$

Therefore $\bar{\mathbf{b}}[\langle U \rangle_m \{M\}] (x w_{\sigma i})(z_j w_{\sigma j})$ for $j \neq i$.

By the induction hypothesis, we have $\hat{\mathbf{b}}[\langle U \rangle_n \{N\}] (x w_{\sigma i})(z_j w_{\sigma j})$. Because we can choose σ arbitrarily, for each $i, j \in \{1, \dots, n\}$ we have $\hat{\mathbf{b}}[\langle U \rangle_n \{N\}] (x w_i)(x w_j)$. Hence we obtain our first target:

$$\langle U \rangle_n \{N\} (x w_1) \dots (x w_n).$$

Next we will show

$$\forall z_1 \dots z_{i-1} z_{i+1} \dots z_n . \langle T \to U \rangle_n \{N\} z_1 \dots z_{i-1} x z_{i+1} \dots z_n \supset\subset \langle T \to U \rangle_m \{N\} z_1 \dots z_{i-1} y z_{i+1} \dots z_m .$$

It is enough to show

$$\begin{aligned} &\forall z_1 \dots z_{i-1} z_{i+1} \dots z_n .\ \forall w_1 \dots w_n . \\ &\quad \langle T \rangle_n \{N\} w_1 \dots w_n \supset (\ \langle T \to U \rangle_n \{N\} (z_1 w_1) \dots (x w_i) \dots (z_n w_n) \\ &\qquad\qquad \supset\subset \langle T \to U \rangle_n \{N\} (z_1 w_1) \dots (y w_i) \dots (z_n w_n)\ ). \end{aligned}$$

Suppose that $\langle T \rangle_n \{N\} w_1 \dots w_n$, then we have $\mathbf{D}^m_n[\langle T \rangle_n \{N\}] w_i \dots w_i$.

By the induction hypothesis, $\mathbf{C}^m_m[\langle T \rangle_m \{M\}] w_i \dots w_i$. Thus we have $\langle T \rangle_m \{M\} w_i \dots w_i$.

As the assumption, there are some numbers $j, j' \in \{1, \dots, m\}$ and a sequence $z_1, z_2, \dots$ such that $x = z_j \wedge y = z_{j'} \wedge \langle T \to U \rangle_m \{M\} z_1 \dots z_m$. Thus

$$x w_i = z_j w_i \wedge y = z_{j'} w_i \wedge \langle U \rangle_m \{M\} (z_1 w_i) \dots (z_m w_i),$$

Therefore $\bar{\mathbf{b}}[\langle U \rangle_m \{M\}] (x w_i)(y w_i)$.

By the induction hypothesis, we have $\hat{\mathbf{b}}[\langle U \rangle_n \{N\}] (x w_i)(y w_i)$. Hence we obtain the second target:

$$\forall z_1 \dots z_{i-1} z_{i+1} \dots z_n . \langle T \to U \rangle_n \{N\} z_1 \dots z_{i-1} x z_{i+1} \dots z_n \supset\subset \langle T \to U \rangle_m \{N\} z_1 \dots z_{i-1} y z_{i+1} \dots z_m . \qquad \square$$

4.6. Proof of the Theorem

Proof of Theorem 4.6:

For each type $T \in \mathsf{R}$, we have

$$\mathbf{Par}_m(\mathsf{R}) \vdash \mathbf{D}^n_m[\langle T \rangle^=_m] \sqsubseteq \mathbf{C}^n_n[\langle T \rangle^=_n],$$

$$\mathbf{Par}_m(\mathsf{R}) \vdash \mathbf{D}^n_n[\langle T \rangle^=_n] \sqsubseteq \mathbf{C}^n_m[\langle T \rangle^=_m]$$

by Lemma 4.37. $\mathbf{Par}_m(\mathsf{R})$ also proves $\mathcal{E}q^m_T \cong \langle T \rangle^=_m$. $\mathbf{L}$ derives $\mathcal{E}q^n_T \cong \mathbf{D}^n_m[\mathcal{E}q^m_T]$, and $\mathcal{E}q^n_T \cong \mathbf{C}^n_m[\mathcal{E}q^m_T]$. Thus we obtain the theorem. $\square$

5. Realizability Interpretation

In this section we shall discuss the realizability interpretation for **L**, and we shall show the self-realizability of **Par**(R). By using this self-realizability, we shall obtain the term extraction property of **Par**(R).

5.1. Realizability of the Basic Logic

First we define the realizability interpretation of **L** into **L** itself.

Definition 5.1. A figure $\{(\Phi_1, \Psi_1)/\phi_1, \ldots, (\Phi_n, \Psi_n)/\phi_n\}$ denotes the assignment which assigns predicates (Φ_i, Ψ_i) to ϕ_i, where the predicate type of Φ_i is A, which is the same as the predicate type of ϕ_i, and the predicate type of Ψ_i is $T \to A$ for some type T. It is called a *double assignment*.

Definition 5.2. For a double assignment $\Xi \equiv \{(\Phi_1, \Psi_1)/\phi_1, \ldots, (\Phi_n, \Psi_n)/\phi_n\}$, The notation $[\Xi]$ stands for the substitution such that $[\Xi] :\equiv [\Phi_1/\phi_1 \ldots \Phi_n/\phi_n]$.

We want to define $R\{\Xi\} \models P$ for a formula P. For convenience, we shall define $R\{\Xi\} \models \Phi$ for a predicate Φ, not only for formulae. First we define $\mathbf{r}^{\Xi}(\Phi)$. It will be the type of R in $R\{\Xi\} \models \Phi$. The notation $\mathbf{r}^{\Xi}(\Phi)$ depends only on the predicate types of Ψ and Ψ' of $\{(\Phi, \Psi')/\phi\}$ in Ξ.

Definition 5.3. We define a type $\mathbf{r}^{\Xi}(\Phi)$ for a predicate Φ and a double assignment Ξ.
$\mathbf{r}^{\Xi}(\phi) :\equiv T$, if Ξ assigns $(\Phi^A, \Psi^{T \to A})$ to ϕ,
$\mathbf{r}^{\Xi}(\phi) :\equiv \mathbf{1}$, if Ξ does not assign predicates to ϕ,
$\mathbf{r}^{\Xi}(\lambda x.\Phi) :\equiv \mathbf{r}^{\Xi}(\Phi)$,
$\mathbf{r}^{\Xi}(\Phi X) :\equiv \mathbf{r}^{\Xi}(\Phi)$,
$\mathbf{r}^{\Xi}(P \supset Q) :\equiv \mathbf{r}^{\Xi}(P) \to \mathbf{r}^{\Xi}(Q)$,
$\mathbf{r}^{\Xi}(\forall x^T.P) :\equiv T \to \mathbf{r}^{\Xi}(P)$,
$\mathbf{r}^{\Xi}(\forall \gamma.P) :\equiv \Pi\gamma.\mathbf{r}^{\Xi}(P)$,
$\mathbf{r}^{\Xi}(\forall \phi^A.P) :\equiv \Pi\gamma.\mathbf{r}^{\Xi'}(P)$, where $\Xi' \equiv \{(\phi, \psi)/\phi, \Xi\}$ for a fresh $\psi^{\gamma \to A}$.

Definition 5.4. For a predicate Φ, a double assignment Ξ and a term R of type $\mathbf{r}^{\Xi}(\Phi)$, we define the realizability interpretation $R\{\Xi\} \models \Phi$. It is defined by induction on Φ.
$R\{\Xi\} \models \phi :\equiv \lambda x_1 \ldots x_n.(\Psi R x_1 \ldots x_n \wedge \Phi x_1 \ldots x_n)$, if Ξ assigns (Φ, Ψ) to ϕ,
$R\{\Xi\} \models \phi :\equiv \phi$, if Ξ does not assign predicates to ϕ,
$R\{\Xi\} \models \lambda x.\Phi :\equiv \lambda x.(R\{\Xi\} \models \Phi)$,
$R\{\Xi\} \models \Phi X :\equiv (R\{\Xi\} \models \Phi)X$,
$R\{\Xi\} \models P \supset Q :\equiv (\forall r.(r\{\Xi\} \models P) \supset (Rr\{\Xi\} \models Q)) \wedge (P \supset Q)[\Xi]$,
$R\{\Xi\} \models \forall x.P :\equiv \forall x.Rx\{\Xi\} \models P$,
$R\{\Xi\} \models \forall \gamma.P :\equiv \forall \gamma.R\gamma\{\Xi\} \models P$,
$R\{\Xi\} \models \forall \phi^A.P :\equiv \forall \gamma.\forall \phi^A \psi^{\gamma \to A}.R\gamma\{(\phi, \psi)/\phi, \Xi\} \models P$.

In what follows, $\mathbf{r}(\Phi)$ stands for $\mathbf{r}^{\{\ \}}(\Phi)$ and $R \models P$ stands for $R\{\} \models P$.

Proposition 5.5. If P has no free predicate variables, then $R \models P \ \equiv\ R\{\Xi\} \models P$.

Definition 5.6. For a double assignment $\Xi \equiv \{(\Phi_1, \Psi_1)/\phi_1, \ldots, (\Phi_n, \Psi_n)/\phi_n\}$,

$$Env(\Xi) :\equiv (\bigcup_i Env(\Phi_i)\ ) \cup (\bigcup_i Env(\Psi_i)\ ).$$

Lemma 5.7. For each predicate Φ of predicate type $T_1 \to \cdots T_n \to *$ and each assignment Ξ, **L** derives

$$Env(\Phi, R, \Xi), x_1 : T_1, \ldots, x_n : T_n, (R\{\Xi\} \models \Phi)x_1 \ldots x_n \vdash \Phi[\Xi]x_1 \ldots x_n.$$

Proof:
By induction on Φ. $\square$

Corollary 5.8. (Completeness) For each formula P and each assignment Ξ, $\mathbf{L}$ derives

$$Env(P, R, \Xi), (R\{\Xi\} \models P) \vdash P[\Xi].$$

Proof:
Immediately from Lemma 5.7. $\square$

Lemma 5.9. If $E, Q_1, \ldots, Q_n \vdash P$ then there is a term R such that $Env(R) \subset E$ and $\mathbf{L}$ derives $E \vdash (R \models Q_1 \supset \cdots \supset Q_n \supset P)$.

Proof:
By induction on the derivation of $E, Q_1, \ldots, Q_n \vdash P$. $\square$

Corollary 5.10. (Soundness) If $\vdash P$ then there is a closed term R such that $\mathbf{L}$ derives $\vdash (R \models P)$.

Proof:
Immediately from 5.9. $\square$

As a corollary of completeness, we have the term extraction property for realizable formulae.

Lemma 5.11. Let $\mathbf{A}$ be a set of axioms. If $\mathbf{A} \vdash (R \models \forall x.\exists y.P)$, then there is a term X such that $\mathbf{A} \vdash \forall x.P[Xx/y]$.

Proof:
There is closed terms X and Y such that $\mathbf{A} \vdash \forall x.(Yx \models P[Xx/y])$. We also have $x \vdash (Yx \models P[Xx/y]) \supset P[Xx/y]$ by the completeness (Corollary 5.8). $\square$

Lemma 5.12. Let $\mathbf{A}$ be a set of axioms such that for each $A \in \mathbf{A}$ there is a term R such that $\mathbf{A} \vdash (R \models A)$. If $\mathbf{A} \vdash P$, then there is a close term R such that $\mathbf{A} \vdash (R \models P)$.

Proof:
Immediately from the soundness (Corollary 5.10). $\square$

Corollary 5.13. Let $\mathbf{A}$ be a set of axioms. Suppose that for each $A \in \mathbf{A}$ there is a term R such that $\mathbf{A} \vdash (R \models A)$. If $\mathbf{A} \vdash \forall x.\exists y.P$, then there is a term X such that $\mathbf{A} \vdash \forall x.P[Xx/y]$.

Proof:
Immediately from Lemmata 5.11 and 5.12. $\square$

We call $\mathbf{A}$ *self-realizable* when there is a term R_A such that $\mathbf{A} \vdash (R_A \models A)$ for each $A \in \mathbf{A}$. If $\mathbf{A}$ is self-realizable, the theory of $\mathbf{A}$ satisfies term extraction.

We shall show that $\mathbf{Par}(\mathsf{R})$ is self-realizable. What we must do is to show that for each T in R there is a term R_T such that $\mathbf{Par}(\mathsf{R}) \vdash R_T \models \mathbf{Conf}(T)$, which is equivalent to $\mathbf{Par}(\mathsf{R}), x^T \vdash R_T x \models \langle T\rangle^= xx$. For the purpose of showing that, we shall first define a predicate $\langle T\rangle^{\models}$ for each T, and shall show that $\mathbf{Par}(\mathsf{R}) \vdash \forall x^T.\langle T\rangle^{\models} xx(K_T x)$ for some term K_T.

5.2. Conformity for Realizability

Definition 5.14. Let $A_1, \dots, A_n, B_1, \dots, B_n, T_1, \dots, T_n$ be types. Then, a figure

$$\{(A_1, B_1, T_1)/\gamma_1, \dots, (A_n, B_n, T_n)/\gamma_n\}$$

denotes the assignment which assigns a triplet of types (A_i, B_i, T_i) to a type variable γ_i for $1 \le i \le n$. It is called a *triple assignment* .

Definition 5.15. For a triple assignment $\xi \equiv \{(A_1, B_1, T_1)/\gamma_1, \dots, (A_n, B_n, T_n)/\gamma_n\}$, The notations $[\xi]_1$ and $[\xi]_2$ are substitutions such that $[\xi]_1 :\equiv [A_1/\gamma_1, \dots, A_n/\gamma_n]$, and $[\xi]_2 :\equiv [B_1/\gamma_1, \dots, B_n/\gamma_n]$.

Definition 5.16. For a type T and a triple assignment ξ, A type $\mathbf{r}^\xi(T)$ is defined by induction on T as follows:
$\mathbf{r}^\xi(\gamma) :\equiv T$ if ξ assigns (A, B, T) for γ,
$\mathbf{r}^\xi(\gamma) :\equiv \gamma$ if ξ assigns nothing for γ,
$\mathbf{r}^\xi(T \to U) :\equiv T[\xi]_1 \to T[\xi]_2 \to \mathbf{r}^\xi(T) \to \mathbf{r}^\xi(U)$,
$\mathbf{r}^\xi(\Pi\gamma.T) :\equiv \Pi\alpha\beta\delta.\mathbf{r}^{\xi'}(T)$, where $\xi' \equiv \{(\alpha, \beta, \delta)/\gamma, \xi\}$ for fresh type variables α, β, δ.
We write $\mathbf{r}(T)$ for $\mathbf{r}^{\{\ \}}(T)$.

Proposition 5.17. $\mathbf{r}^\xi(T[U/\gamma]) \equiv \mathbf{r}^{\xi'}(T)$ where $\xi' \equiv \{(U[\xi]_1, U[\xi]_2, \mathbf{r}^\xi(U))/\gamma, \xi\}$.

Proof:
By induction on T. □

Definition 5.18. Let $\gamma_1, \dots, \gamma_n$ be free variables, and $\delta_1, \dots, \delta_m$ be all the free variables in T other than $\gamma_1, \dots, \gamma_n$. Let Θ be an assignment such that $\Theta \equiv \{\Phi_1/\gamma_1, \dots, \Phi_n/\gamma_n\}$, where Φ_i is of predicate type $T_i \to T_i \to U_i \to *$. Let ξ be a triple assignment $\{(T_1, T_1, U_1)/\gamma_1, \dots, (T_n, T_n, U_n)/\gamma_n\}$. Then define a predicate

$$\langle T\rangle^{\models}\{\Theta\} :\equiv \lambda xyr.(r\{\Xi\} \models \langle T\rangle\{\phi_1/\gamma_1, \dots, \phi_n/\gamma_n, \psi_1/\delta_1, \dots, \psi_m/\delta_m\}xy)$$

of predicate type $T[\xi]_1 \to T[\xi]_2 \to \mathbf{r}^\xi(T) \to *$, where $\phi_1^{T_1 \to T_1 \to *}, \dots, \phi_n^{T_n \to T_n \to *}, \psi_1^{\delta_1 \to \delta_1 \to *}, \dots, \psi_m^{\delta_m \to \delta_m \to *}$ are fresh predicate variables, and Ξ is a double assignment such that

$$\Xi \equiv \{(\mathcal{E}q_{T_1}, \lambda r_1 x_1 y_1.\Phi_1 x_1 y_1 r_1)/\phi_1, \dots, (\mathcal{E}q_{T_n}, \lambda r_n x_n y_n.\Phi_n x_n y_n r_n)/\phi_n, (\mathcal{E}q_{\delta_1}, \mathcal{E}q^3_{\delta_1})/\psi_1, \dots, (\mathcal{E}q_{\delta_m}, \mathcal{E}q^3_{\delta_m})/\psi_m\}.$$

And $\langle T\rangle^{\models}$ is written for $\langle T\rangle^{\models}\{\ \}$.

For example,
$\langle\gamma\rangle^{\models}xyr =_\beta (\mathcal{E}qxyr \wedge x = y)$,
$\langle\gamma\rangle^{\models}\{\Phi/\gamma\}xyr =_\beta (\wedge\Phi xyr \wedge x = y)$,
$\langle T \to U\rangle^{\models}\{\Theta\}xyr =_\beta \forall vw.(\forall q.\langle T\rangle^{\models}\{v\}wq \supset \langle U\rangle^{\models}\{\Theta\}(xv)(yw)(rq))$
$\wedge(\langle T\rangle^{=}\theta vw \supset \langle U\rangle^{=}\theta(xv)(yw))$
for some substitution of types θ.

5.3. Operations for Functors and Fixed Point Types

Definition 5.19. (Functor's Operation for Realizability) Let F be a functor or a contravariant functor. Let a term X be a term of type $T_1 \to T_2$, a term Y be a term of type $U_1 \to U_2$, and a term R be a term of type $V_1 \to V_2$. If F is a functor, then $\langle\!\langle X, Y, R\rangle\!\rangle_F^\xi$ is a term of type $\mathbf{r}^{\xi_1}(F\gamma) \to \mathbf{r}^{\xi_2}(F\gamma)$, and if F is a contravariant functor, then $\langle\!\langle X, Y, R\rangle\!\rangle_F^\xi$ is a term of type $\mathbf{r}^{\xi_2}(F\gamma) \to \mathbf{r}^{\xi_1}(F\gamma)$, where $\xi_1 \equiv \{(T_1, U_1, V_1)/\gamma, \xi\}$ and $\xi_2 \equiv \{(T_2, U_2, V_2)/\gamma, \xi\}$. They are defined by induction on F. If $F\gamma$ has no occurrence of γ, then $\langle\!\langle X, Y, R\rangle\!\rangle_F^\xi :\equiv \mathbf{I}_{F\gamma}$. If $F\gamma$ has some occurrences of γ, then

- When $F\gamma \equiv \gamma$, then $\langle\!\langle X,Y,R\rangle\!\rangle^{\xi}_{F} :\equiv R$.
- When $F\gamma \equiv G\gamma \to H\gamma$, then
  $\langle\!\langle X,Y,R\rangle\!\rangle^{\xi}_{F} :\equiv \langle\!\langle X,Y,R\rangle\!\rangle^{\xi}_{H}(r(\langle X\rangle_{G}x)(\langle Y\rangle_{G}y)(\langle\!\langle X,Y,R\rangle\!\rangle^{\xi}_{G}q))$.
- When $F\gamma \equiv \Pi\delta.C[\gamma,\delta]$, then $\langle\!\langle X,Y,R\rangle\!\rangle^{\xi}_{F} :\equiv \lambda r.\Lambda\alpha\beta\delta.\langle\!\langle X,Y,R\rangle\!\rangle^{\xi'}_{C[-,\delta]}(r\alpha\beta\delta)$,
  where $\xi' :\equiv \{(\alpha,\beta,\delta)/\delta,\xi\}$.

We write $\langle\!\langle X,Y,R\rangle\!\rangle^{\models}_{F}$ for $\langle\!\langle X,Y,R\rangle\!\rangle^{\{\,\}}_{F}$.

Definition 5.20. (Functor's Operation on Predicate for Realizability) Let F be a functor or a contravariant functor. Let $\Phi^{T\to T\to U\to *}$ be a predicate. Let Θ be an assignment of predicates to type variables. Then $F^{\models}_{\Theta}(\Phi) :\equiv \langle F\gamma\rangle^{\models}\{\Phi/\gamma,\Theta\}$.

Definition 5.21. $\Phi\circ(X,Y,Z) :\equiv \lambda xyz.\Phi(Xx)(Yy)(Zz)$.

As compared with the notation $\circ_i$, it holds that $\Phi\circ(X,Y,Z) =_\beta \Phi\circ_1 X\circ_2 Y\circ_3 Z$.

Proposition 5.22. (Functor Lemma) Let Θ be an assignment such that $\Theta\equiv\{\Phi_1/\gamma_1,\ldots,\Phi_n/\gamma_n\}$, where Φ_i is a predicate of predicate type $T_i\to T_i\to U_i\to *$ for $1\le i\le n$. Let ξ is a triple assignment such that $\xi\equiv\{(T_1,T_1,U_1)/\gamma_1,\ldots,(T_n,T_n,U_n)/\gamma_n\}$. Let F be a functor and G be a contravariant functor. Then $\mathbf{L}$ derives that:

$$Env(X,Y,R,\Phi,\Psi,\Theta),\quad \Phi\sqsubseteq\Psi\circ(X,Y,R)\vdash F^{\models}_{\Theta}\Phi\sqsubseteq F^{\models}_{\Theta}\Psi\circ(\langle X\rangle_{F[\xi]_1},\langle Y\rangle_{F[\xi]_2},\langle\!\langle X,X,R\rangle\!\rangle^{\xi}_{F})$$
$$Env(X,Y,R,\Phi,\Psi,\Theta),\quad \Phi\sqsubseteq\Psi\circ(X,Y,R)\vdash G^{\models}_{\Theta}\Psi\sqsubseteq G^{\models}_{\Theta}\Phi\circ(\langle X\rangle_{G[\xi]_1},\langle Y\rangle_{G[\xi]_2},\langle\!\langle X,X,R\rangle\!\rangle^{\xi}_{G})$$

Proof:
By mutual induction in F and G. $\square$

Definition 5.23.

$\overline{\boldsymbol{c}}^{\models}_{\mu F,T,\xi} :\equiv \lambda xyrq.q(T[\xi_1])(T[\xi]_2)\mathbf{r}^{\xi}(T)xyr$,
of type
$(FT\to T)[\xi]_1\to(FT\to T)[\xi]_2\to\mathbf{r}^{\xi}(FT\to T)\to\mathbf{r}^{\xi}(\mu F)\to\mathbf{r}^{\xi}(T)$,

$\boldsymbol{c}^{\models}_{\mu F,T,\xi} :\equiv \lambda xyrvw.\overline{\boldsymbol{c}}^{\models}_{\mu F,T,\xi}xyr$,
of type
$\mathbf{r}^{\xi}(\,(FT\to T)\to\mu F\to T\,)$,

$\boldsymbol{f}^{\models}_{\mu F,\xi} :\equiv \lambda xyr\Lambda\alpha\beta\gamma.\lambda vwq.$
$q(\langle\boldsymbol{c}_{\mu F[\xi]_1,\alpha}v\rangle_{F[\xi]_1}x)(\langle\boldsymbol{c}_{\mu F[\xi]_2,\beta}w\rangle_{F[\xi]_2}y)(\langle\!\langle\boldsymbol{c}_{\mu F[\xi]_1,\alpha}v,\boldsymbol{c}_{\mu F[\xi]_2,\beta}w,\overline{\boldsymbol{c}}^{\models}_{\mu F,\gamma,\xi'}vwq\rangle\!\rangle^{\xi}_{F}r)$
of type
$\mathbf{r}^{\xi}(F\mu F\to\mu F)$, where $\xi'\equiv\{(\alpha.\beta\gamma)/\gamma,\xi\}$,

$\overline{\boldsymbol{g}}^{\models}_{\mu F,\xi} :\equiv \overline{\boldsymbol{c}}^{\models}_{\mu F,F\mu F,\xi}\langle\boldsymbol{f}_{\mu F[\xi]_1}\rangle_{F[\xi]_1}\langle\boldsymbol{f}_{\mu F[\xi]_2}\rangle_{F[\xi]_2}\lambda xy.\langle\!\langle\boldsymbol{f}_{\mu F[\xi]_1},\boldsymbol{f}_{\mu F[\xi]_2},\boldsymbol{f}^{\models}_{\mu F,\xi}\rangle\!\rangle^{\xi}_{F}$
of type $\mathbf{r}^{\xi}(\mu F)\to\mathbf{r}^{\xi}(F\mu F)$,

$\boldsymbol{g}^{\models}_{\mu F,\xi} :\equiv \lambda xy.\overline{\boldsymbol{g}}^{\models}_{\mu F,\xi}$ of type $\mathbf{r}^{\xi}(\mu F\to F\mu F)$,

$\boldsymbol{c}^{\models}_{\nu F,T,\xi} :\equiv \lambda xyrvwq\Lambda\alpha\beta\gamma.\lambda tup.p(T[\xi_1])(T[\xi]_2)\mathbf{r}^{\xi}(T)xyrvwq$
of type
$\mathbf{r}^{\xi}(\,(T\to FT)\to T\to\nu F\,)$,

$\overline{\boldsymbol{g}}^{\models}_{\nu F,\xi} :\equiv \lambda r.r(F\nu F[\xi]_1)(F\nu F[\xi]_2)\mathbf{r}^{\xi}(F\nu F)\Lambda\alpha\beta\gamma.\lambda vwpxyq.$
$\langle\!\langle\boldsymbol{c}_{\nu F[\xi]_1,\alpha}v,\boldsymbol{c}_{\nu F[\xi]_2,\beta}w,\boldsymbol{c}^{\models}_{\nu F,\gamma,\xi'}vwp\rangle\!\rangle^{\xi}_{F}(pxyq)$
of type
$\mathbf{r}^{\xi}(\nu F)\to\mathbf{r}^{\xi}(F\nu F)$, where $\xi'\equiv\{(\alpha,\beta,\gamma)/\gamma,\xi\}$,

$\boldsymbol{g}^{\models}_{\nu F,\xi} :\equiv \lambda xy.\overline{\boldsymbol{g}}^{\models}_{\nu F,\xi}$
of type
$\mathbf{r}^{\xi}(\nu F\to F\nu F)$,

$\boldsymbol{f}^{\models}_{\nu F,\xi} :\equiv \boldsymbol{c}^{\models}_{\nu F, F\nu F,\xi}\langle \boldsymbol{g}_{\nu F[\xi]_1}\rangle_{F[\xi]_1}\langle \boldsymbol{g}_{\nu F[\xi]_2}\rangle_{F[\xi]_2}\lambda xy.\langle\!\langle \boldsymbol{g}_{\nu F[\xi]_1}, \boldsymbol{g}_{\nu F[\xi]_2}, \boldsymbol{g}^{\models}_{\nu F,\xi}\rangle\!\rangle^{\xi}_{F}$
of type
$\mathbf{r}^{\xi}(F\nu F \to \nu F)$.

We shall often omit the suffix ξ when it is detectable. Moreover, we shall sometimes omit the type suffixes, namely, $\boldsymbol{c}^{\models}_{T}$ and $\boldsymbol{c}^{\models}$ are written for $\boldsymbol{c}^{\models}_{\mu F,T,\xi}$ for some ξ.

Proposition 5.24. $\mathbf{L}$ derives the following:
$Env(\Theta) \vdash \langle (FT \to T) \to \mu F \to T\rangle^{\models}\{\Theta\}\boldsymbol{ccc}^{\models}$,
$Env(\Theta) \vdash \langle F\mu F \to \mu F\rangle^{\models}\{\Theta\}\boldsymbol{fff}^{\models}$,
$Env(\Theta) \vdash \langle \mu F \to F\mu F\rangle^{\models}\{\Theta\}\boldsymbol{ggg}^{\models}$,
$Env(\Theta) \vdash \langle (T \to FT) \to T \to \nu F\rangle^{\models}\{\Theta\}\boldsymbol{ccc}^{\models}$,
$Env(\Theta) \vdash \langle \nu F \to F\nu F\rangle^{\models}\{\Theta\}\boldsymbol{ggg}^{\models}$,
$Env(\Theta) \vdash \langle F\nu F \to \nu F\rangle^{\models}\{\Theta\}\boldsymbol{fff}^{\models}$.

Proposition 5.25. $\mathbf{L}$ derives the following:
$Env(X,Y,R,\Theta), \langle FT \to T\rangle^{\models}\{\Theta\}XYR \vdash$
$\langle \mu F\rangle^{\models}\{\Theta\} \sqsubseteq \langle T\rangle^{\models}\{\Theta\} \circ (\boldsymbol{c}X, \boldsymbol{c}Y, \overline{\boldsymbol{c}}^{\models}XYR)$,
$Env(\Theta) \vdash \langle \mu F\rangle^{\models}\{\Theta\} \sqsubseteq \langle F\mu F\rangle^{\models}\{\Theta\} \circ (\boldsymbol{g}, \boldsymbol{g}, \overline{\boldsymbol{g}}^{\models})$,
$Env(\Theta) \vdash \langle \nu F\rangle^{\models}\{\Theta\} \sqsubseteq \langle F\nu F\rangle^{\models}\{\Theta\} \circ (\boldsymbol{g}, \boldsymbol{g}, \overline{\boldsymbol{g}}^{\models})$.

5.4. Crucial Lemma

Definition 5.26. For each type $T \in \mathsf{R}$, terms $\mathbf{J}_T$ of type $\mathbf{r}(T) \to T$ and $\mathbf{K}_T$ of type $T \to \mathbf{r}(T)$ are defined by mutual induction on T as below:
$\mathbf{J}_\gamma \equiv \mathbf{K}_\gamma :\equiv \mathbf{I}_\gamma$,
$\mathbf{J}_{\mathbf{1}} :\equiv \lambda r.\Lambda\gamma.\mathbf{I}_\gamma$,
$\mathbf{K}_{\mathbf{1}} :\equiv \lambda x.\Lambda\alpha\beta\gamma.\lambda y^{\alpha}z^{\beta}r^{\gamma}.r$,
$\mathbf{J}_{T\to U} :\equiv \lambda rx.\mathbf{J}_U(rxx(\mathbf{K}_T x))$,
$\mathbf{K}_{T\to U} :\equiv \lambda xyzr.\mathbf{K}_U(x(\mathbf{J}_T r))$,
$\mathbf{J}_{T\times U} :\equiv \lambda r\Lambda\gamma.\mathbf{J}_{(T\to U\to\gamma)\to\gamma}(r\gamma\gamma\gamma)$,
$\mathbf{K}_{T\times U} :\equiv \lambda x\Lambda\alpha\beta\gamma.\mathbf{K}_{(T\to U\to\gamma)\to\gamma}(x\gamma)$,
$\mathbf{J}_{T+U} :\equiv \lambda r\Lambda\gamma.\mathbf{J}_{(T\to\gamma)\to(U\to\gamma)\to\gamma}(r\gamma\gamma\gamma)$,
$\mathbf{K}_{T+U} :\equiv \lambda x\Lambda\alpha\beta\gamma.\mathbf{K}_{(T\to\gamma)\to(U\to\gamma)\to\gamma}(x\gamma)$,
$\mathbf{J}_{\mu F} :\equiv \overline{\boldsymbol{c}}^{\models}_{\gamma}[\mu F/\gamma]\boldsymbol{ff}\lambda xyr.\boldsymbol{f}(\mathbf{J}_{F\gamma}[\mu F/\gamma]r)$,
$\mathbf{K}_{\mu F} :\equiv \boldsymbol{c}_{\mathbf{r}(\mu F)}\lambda x.\boldsymbol{f}^{\models}(\langle \mathbf{J}_{\mu F}\rangle_F x)(\langle \mathbf{J}_{\mu F}\rangle_F x)(\mathbf{K}_{F\gamma}[\mathbf{r}(\mu F)/\gamma]x)$,
$\mathbf{J}_{\nu F} :\equiv \boldsymbol{c}_{\mathbf{r}(\nu F)}(\mathbf{J}_{F\gamma}[\mathbf{r}(\nu F)/\gamma] \circ \overline{\boldsymbol{g}}^{\models})$,
$\mathbf{K}_{\nu F} :\equiv \lambda x.\boldsymbol{c}^{\models}_{\gamma}[\nu F/\gamma]\boldsymbol{gg}(\mathbf{K}_{F\gamma}[\nu F/\gamma] \circ \boldsymbol{g})xxx$.

Proposition 5.27. $\mathbf{Par}(\mathsf{R}) \vdash \mathbf{I}_T = \mathbf{J}_T \circ \mathbf{K}_T$.

Definition 5.28. For a predicate Φ of predicate type $T \to T \to \mathbf{r}(T) \to *$, we define predicates $\mathbf{C}^{\models}[\Phi]$ and $\mathbf{D}^{\models}[\Phi]$ of predicate type $T^3 \to *$.
$\mathbf{C}^{\models}[\Phi] :\equiv \lambda xyz.\mathcal{E}qxyz \wedge \Phi xy(K_T z)$,
$\mathbf{D}^{\models}[\Phi] :\equiv \lambda xyz.\mathcal{E}qxyz \vee \exists r^{\mathbf{r}(T)}.z = J_T r \wedge \Phi xyr$.

Definition 5.29. For a predicate Φ of predicate type $T^3 \to *$, we define predicates $\mathbf{C}'[\Phi]$ and $\mathbf{D}'[\Phi]$ of the same predicate type $T^3 \to *$.
$\mathbf{C}'[\Phi] :\equiv \lambda x^T y^T z^T.\mathcal{E}qxyz \wedge \Phi xyz$,
$\mathbf{D}'[\Phi] :\equiv \lambda x^T y^T z^T.\mathcal{E}qxyz \vee \Phi xyz$.

Definition 5.30. For an assignment $\Theta \equiv \{\Phi/\gamma, \Psi/\delta, \ldots\}$,
$\mathbf{C}^{\models}[\Theta] :\equiv \{\mathbf{C}^{\models}[\Phi]/\gamma, \mathbf{C}^{\models}[\Psi]/\delta, \ldots\}$,
$\mathbf{D}^{\models}[\Theta] :\equiv \{\mathbf{D}^{\models}[\Phi]/\gamma, \mathbf{D}^{\models}[\Psi]/\delta, \ldots\}$.

Definition 5.31. For an assignment $\Theta \equiv \{\Phi/\gamma, \Psi/\delta, \ldots\}$,
$\mathbf{C}'[\Theta] :\equiv \{\mathbf{C}'[\Phi]/\gamma, \mathbf{C}'[\Psi]/\delta, \ldots\}$,
$\mathbf{D}'[\Theta] :\equiv \{\mathbf{D}'[\Phi]/\gamma, \mathbf{D}'[\Psi]/\delta, \ldots\}$.

Lemma 5.32. (Crucial Lemma) Let T, $T_1, \ldots, T_i$, $T'_1, \ldots, T'_{i'}$, $U_1, \ldots, U_j$, $U'_1, \ldots, U'_{j'}$ be types in R. Suppose that type variables $\gamma_1, \ldots, \gamma_i, \gamma'_1, \ldots, \gamma'_{i'}$ occur only positively in T, and $\delta_1, \ldots, \delta_j, \delta'_1, \ldots, \delta'_{j'}$ occur only negatively in T. Let

$$\Phi_1{}^{T_1 \to T_1 \to \mathbf{r}(T_1) \to *}, \ldots, \Phi_i{}^{T_i \to T_i \to \mathbf{r}(T_i) \to *}, \Phi'_1{}^{T'^3_1 \to *}, \ldots, \Phi'_{i'}{}^{T'^3_{i'} \to *},$$

$$\Psi_1{}^{U_1 \to U_1 \to \mathbf{r}(U_1) \to *}, \ldots, \Psi_j{}^{U_j \to U_j \to \mathbf{r}(U_j) \to *}, \Psi'_1{}^{U'^3_1 \to *}, \ldots, \Psi'_{j'}{}^{U'^3_{j'} \to *}$$

be predicates. Let assignments Θ, Θ', Ξ and Ξ' be defined as:

$$\Theta :\equiv \{\Phi_1/\gamma_1, \ldots, \Phi_i/\gamma_i\}, \quad \Theta' :\equiv \{\Phi'_1/\gamma'_1, \ldots, \Phi'_{i'}/\gamma'_{i'}\},$$

$$\Xi :\equiv \{\Psi_1/\delta_1, \ldots, \Psi_j/\delta_j\}, \ \ \Xi' :\equiv \{\Psi'_1/\delta'_1, \ldots, \Psi'_{j'}/\delta'_{j'}\}.$$

Let θ and θ' be substitutions of types such as

$$\theta \equiv [T_1/\gamma_1, \ldots, T_i/\gamma_i, U_1/\delta_1, \ldots, U_j/\delta_j], \quad \theta \equiv [T'_1/\gamma'_1, \ldots, T'_{i'}/\gamma'_{i'}, U'_1/\delta'_1, \ldots, U'_{j'}/\delta'_{j'}].$$

Then
$Env(\Theta, \Theta', \Xi, \Xi'), \mathbf{Par}(\mathsf{R}) \vdash$

$$\langle T \rangle_3 \{\mathbf{C}^{\models}[\Theta], \mathbf{C}'[\Theta'], \mathbf{D}^{\models}[\Xi], \mathbf{D}'[\Xi']\} \sqsubseteq \langle T \rangle^{\models}\{\Theta, \Theta', \Xi, \Xi'\} \circ_3 \mathbf{K}_{T\theta}\theta',$$

$Env(\Theta, \Theta', \Xi, \Xi'), \mathbf{Par}(\mathsf{R}) \vdash$

$$\langle T \rangle^{\models}\{\Theta, \Theta', \Xi, \Xi'\} \sqsubseteq \langle T \rangle_3 \{\mathbf{D}^{\models}[\Theta], \mathbf{D}'[\Theta'], \mathbf{C}^{\models}[\Xi], \mathbf{C}'[\Xi']\} \circ_3 \mathbf{J}_{T\theta}\theta'.$$

Proof:
By induction of construction on T. $\Box$

Corollary 5.33. For $T \in \mathsf{R}$, $\mathbf{Par}(\mathsf{R}) \vdash \forall x^T . \langle T \rangle^{\models} xx(\mathbf{K}_T x)$.

Proof:
Immediately from the crucial lemma(Lemma 5.32). $\Box$

5.5. Self-Realizability of Parametricity

Lemma 5.34. (Realizability of Conformity) For a type $T \in \mathsf{R}$, there is a term R such that $\mathbf{Par}(\mathsf{R}) \vdash (R \models \mathbf{Conf}(T))$.

Proof:
The total proof consists of three steps.
As the first step,

$$\mathbf{L} \vdash (\mathcal{E}q_{\gamma_1} \cong \phi_1) \supset \cdots \supset (\mathcal{E}q_{\gamma_n} \cong \phi_n) \supset \langle T \rangle^{=} \cong \langle T \rangle \{\phi_1/\gamma_1, \ldots, \phi_n/\gamma_n\},$$

where ϕ is a predicate variable of predicate type $\gamma_i \to \gamma_i \to *$ for $1 \leq i \leq n$. Thus there is a closed term R_0 such that

$$\mathbf{L} \vdash R_0\{\Xi\} \models (\mathcal{E}q_{\gamma_1} \cong \phi_1) \supset \cdots \supset (\mathcal{E}q_{\gamma_n} \cong \phi_n) \supset \langle T \rangle^{=} \cong \langle T \rangle \{\phi_1/\gamma_1, \ldots, \phi_n/\gamma_n\}.$$

where we can choose an arbitrary double assignment Ξ. We put

$$\Xi :\equiv \{(\mathcal{E}q_{\gamma_1}, \mathcal{E}q^3_{\gamma_1})/\phi_1, \dots (\mathcal{E}q_{\gamma_n}, \mathcal{E}q^3_{\gamma_n})/\phi_n\}.$$

As the next step, there is a closed term R such that $\mathbf{L} \vdash R_i\{(\mathcal{E}q, \mathcal{E}q^3)/\phi_i\} \models \mathcal{E}q \cong \phi_i$. Thus we have

$$\mathbf{L} \vdash R_0 R_1 \dots R_n\{\Xi\} \models \langle T\rangle^= \cong \langle T\rangle\{\phi_1/\gamma_1, \dots, \phi_n/\gamma_n\}.$$

As the last step, by the corollary of the crucial lemma (Corollary 5.33) we have

$$\mathbf{Par}(\mathsf{R}) \vdash \forall x^T.\ \mathbf{K}_T x\{\Xi\} \models \langle T\rangle\{\phi_1/\gamma_1, \dots, \phi_n/\gamma_n\} x x.$$

Thus there is closed term R' such that

$$\mathbf{Par}(\mathsf{R}) \vdash \forall x^T.\ R'(\mathbf{K}_T x)\{\Xi\} \models \langle T\rangle^= x x.$$

Hence we have $\mathbf{Par}(\mathsf{R}) \vdash R' \circ \mathbf{K}_T\{\Xi\} \models \mathbf{Conf}(T)$. By Proposition 5.5, this is identical to $\mathbf{Par}(\mathsf{R}) \vdash R' \circ \mathbf{K}_T \models \mathbf{Conf}(T)$.
□

Lemma 5.35. (Self-Realizability of Extensionality) For each type T, there is a term R such that $\mathbf{Ext}(T) \vdash (R \models \mathbf{Ext}(T))$.

Proof:
By $R \equiv \Lambda\gamma\lambda x^T y^T z r^\gamma.r$. □

By using these lemmata, we obtain the main theorem of this section.

Theorem 5.36. $\mathbf{Par}(\mathsf{R})$ is self-realizable, that is, for each type $T \in \mathsf{R}$, there is a term R such that $\mathbf{Par}(\mathsf{R}) \vdash (R \models \mathbf{Par}(T))$.

Proof:
By realizability of conformity (Lemma 5.34) and self-realizability of extensionality (Lemma 5.35). □

Corollary 5.37. $\mathbf{Par}(\mathsf{R})$ satisfies the term extraction property.

Proof:
By Lemma 5.13 and Theorem 5.36. □

6. Conclusion

This paper has discussed the system of parametricity of Plotkin and Abadi [8]. We have pointed out three new results.

The first is consistency of the parametricity. The second is that the polyadic parametricities for recursive types are equivalent to each other. The third is a sound realizability interpretation of the theory of parametricity for recursive types.

Acknowledgement
I am greatly indebted to M. Sato for his total support for this work. I am deeply indebted to J. R. Hindley for carefully reading the draft of this paper and checking my English expression. I am very indebted to G. D. Plotkin, and P. Dybjer for many previous discussions and comments. I would like to thank very much T. Ito, and M. H. Takahashi for their encouragement and comments. I would like to thank R. Hasegawa, P. O'Hearn, Y. Akama, D. Suzuki, M. Hasegawa, and the referees for discussions and comments.

References

[1] M. Abadi, L. Cardelli and P.-L. Curien. Formal parametric polymorphism. *Theoretical Computer Science,* 121:9–58, 1993.

[2] H. Barendregt. Lambda Calculi with Types. In *Handbook of Logic in Computer Science,* Vol. 2, pages 117–309, Oxford Univ. Press, 1992, edited by S. Abramsky et al.

[3] R. Bellucci, Martin Abadi and P.-L. Curien. A Model for Formal Parametric Polymorphism: A PER Interpretation for system R. In *Typed Lambda Calculi and Applications,* Lecture Notes in Computer Science, 902, pages 32–46, Springer Verlag, 1995, edited by G. Plotkin et al.

[4] J.-Y. Girard et al. *Proofs and Types,* Cambridge Univ. Press, 1989.

[5] R. Harper, F. Honsell and G. Plotkin. A Framework for Defining Logics. *Journal of the Association for Computing Machinery,* 40:143–184, 1993.

[6] R. Hasegawa. Parametricity of extensionally collapsed term models of polymorphism and their categorical properties. In *Proceedings of Theoretical Aspects of Computer Science,* Lecture Notes In Computer Science 526, pages 495–512, Springer-Verlag, 1991, edited by T. Ito et al.

[7] R. Hasegawa. Categorical data types in parametric polymorphism. In *Mathematical Structures in Computer Science,* 4:71–109, 1994.

[8] G. Plotkin and M. Abadi. A Logic for parametric polymorphism. In *Typed Lambda Calculi and Applications,* Lecture Notes in Computer Science, 664, pages 361–375, Springer Verlag, 1993, edited by M. Bezem et al.

[9] J. C. Reynolds. Types, abstraction and parametric polymorphism. In R. E. A. Mason, editor, *Information Processing 83,* pages 513–523, Elsevier Science Publishers B. V., 1983.