Thesis for the Degree of Licentiate of Engineering

Pure Type Systems with an Internalized Parametricity Theorem

Guilhem Moulin

Department of Computer Science and Engineering
Chalmers University of Technology
Göteborg, Sweden 2013

Pure Type Systems with an Internalized Parametricity Theorem
Guilhem Moulin

Technical Report 109L
ISSN 1652-876X
Department of Computer Science and Engineering
Programming Logic Research Group

Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone +46 (0)31-772 10 00

Printed at Chalmers
Göteborg, Sweden 2013

Abstract

Parametricity results have recently been proved for dependently-typed calculi such as the Calculus of Constructions. However these results are meta theorems, and although they can be stated as internal propositions, they cannot be proved internally. In this thesis we define for any sufficiently strong Pure Type System $\mathcal{O}$ (such as the Calculus of Constructions) an extension $\mathcal{P}$ in which each instance of the parametricity theorem, including those corresponding to open terms, can be proved internally. As a consequence we can prove inside the system that each term of type $\forall A.A \to A$ is an identity. Furthermore, our system $\mathcal{P}$ is proved to be strongly normalizing by a reduction-preserving interpretation into $\mathcal{O}$. We also prove Church-Rosser and Subject Reduction properties; consistency follows.

Keywords: Polymorphism, Parametricity, Type structure, Lambda Calculus.

The present thesis is an extended version of the paper *A computational interpretation of parametricity* which appeared in the Proceedings of LICS 2012 [Bernardy and Moulin, 2012].

Contents

Proofs for free **5**
1.1 Pure type systems 5
1.2 Logical relations, from PTS to PTS 7

Towards internalizing parametricity **11**
2.1 Aim and example 11
2.2 Internalization 13
2.3 Parametricity of parametricity 15
2.4 A syntax for hypercubes 17
2.5 The interpretation of hypercubes 19
2.6 Exchanging dimensions 21
2.7 Dimension checks 24

A calculus with an internal parametricity theorem **25**
3.1 Definitions 25
3.2 Properties of the Parametric Interpretation 30
3.3 Confluence 32
3.4 Abstraction 36
3.5 Subject Reduction 38
3.6 Reduction-preserving model into the underlying PTS . . . 39

Bibliography **46**

Appendix: Additional proofs **47**

Acknowledgements

I would like to show my gratitude to my supervisors Peter Dybjer and Jean-Philippe Bernardy for their patience and extremely valuable guidance along the way. Jean-Philippe deserves a special mention since his intuition is to a large extend what gave birth to this work and to our publications in the LICS 2012 and ICFP 2013 proceedings. Many thanks also to Peter for his tireless readings and numerous corrections and other improvements on this thesis.

I am also grateful to Thorsten Altenkirch to have accepted the role of discussion leader for my defense, and to my fellow members of the Proglog research group for the insightful weekly seminars. Thanks to my teachers André Hirschowitz and Yves Bertot for their trust which made my moving to Sweden possible.

Thanks also to my colleagues and students, who make the department such a nice place to work at. Thanks to my officemates and friends Simon Huber, Bassel Mannaa and Anders Mörtberg for the good time and political discussions that will probably never come to an end. Finally, thanks to Nils Anders Danielsson to have kindly lent me the unicycle I am now riding daily, on the way to work.

Introduction

Parametricity, as formally stated by Reynolds [1983], expresses that polymorphic functions must behave *uniformly*. This is done by interpreting a type A as a relation $[\![A]\!] : A \to A \to \star$ such that $[\![A]\!]\, a\, a$ for every $a : A$. In other words this result, known as the *abstraction* theorem, says that every type gives a theorem which holds for any of its inhabitants.

The study of parametricity, starting with Reynolds' work, is typically semantic: originally the abstraction theorem was proved for types of system F, and the concern was to construct a model capturing its polymorphic character. Later Mairson [1991], followed by Abadi et al. [1993], developed a more syntactic approach: types were interpreted in another calculus (of proofs and propositions), and for each proof term, they showed how to construct a proof term inhabiting the relational interpretation of its types.

Bernardy et al. [2010], Bernardy and Lasson [2011] have more recently shown how to extend the relational interpretation to some dependent type theories, such as the Calculus of Constructions [Coquand and Huet, 1986] or Martin-Löf's Intuitionistic Type Theory [1984]. They also show how terms, types and their relational interpretations as proofs and propositions can all be expressed in the same calculus.

For instance, in the Calculus of Constructions, the interpretation of any $f : \forall x : A.\, A \to A$ gives that it must be an identity, in other words that x is Leibniz-equal to $f\, A\, x$ for each $x : A$:

$$\forall A : \star.\, \forall x : A.\, x \equiv_A f\, A\, x \tag{1}$$

where the Leibniz equality $x \equiv_A y$ is defined (for $A : \star$ and $x, y : A$) as $\forall P : (A \to \star).\, P\, x \to P\, y$.

The notion of parametricity, in particular the abstraction theorem, is used in numerous applications when reasoning about functional programs [Wadler, 1989], for instance to prove the correctness of shortcut fusion [Gill et al., 1993, Johann, 2002]. Relying on parametricity

conditions is also required when using Church-encoding to represent datatypes [Plotkin and Abadi, 1993].

Parametricity theorems have also been used for richer calculi, *e.g.*, the Calculus of Inductive Constructions [Pfenning and Paulin-Mohring, 1990, Keller and Lasson, 2012], for instance to prove the correctness of well-scoped representations of λ-terms [Chlipala, 2008, Pouillard, 2011]. Indeed, an informal justification of the fact that all inhabitants of the following inductive definition (here in Agda syntax, and due to Pouillard [2011]) are well-scoped lies in the fact that *the index V is abstract*, hence the only way to introduce new variables is by abstraction.

$$
\begin{array}{l}
\textbf{data}\ \mathsf{Term}\ (V : \star) : \star\ \textbf{where} \\
\quad \mathsf{var}\ : V \to \mathsf{Term}\ V \\
\quad \mathsf{app} : \mathsf{Term}\ V \to \mathsf{Term}\ V \to \mathsf{Term}\ V \\
\quad \mathsf{abs}\ : \mathsf{Term}\ (\mathsf{Maybe}\ V) \to \mathsf{Term}\ V
\end{array}
$$

However, the *parametricity property*, *i.e.*, the fact that every term satisfies the parametric interpretation of its type, has not been known to be provable in the system in which the type is expressed. In particular, the following property cannot be proved in the Calculus of Constructions or Martin-Löf's Intuitionistic Type Theory, hence cannot be proved either in existing proof assistants based on these systems, such as Coq [The Coq development team, 2013] or Agda [Norell, 2007].

$$
\begin{array}{l}
f : \forall (A : \star).\, A \to A \\
A : \star \\
x : A \\
\hline
f\ A\ x \equiv_A x
\end{array}
$$

(Were $\equiv$ stands for the Leibnitz equality.)

On the other hand, one may notice that the parametricity condition associated with the polymorphic identity is the missing assumption to prove this property.

In fact, users relying on the parametricity conditions have postulated the parametricity axiom [Pouillard, 2011, Chlipala, 2008, Atkey et al., 2009]. However, this approach has a fundamental drawback: because the postulate does not have a computational interpretation, parametricity conditions can only be used in computationally-irrelevant positions. Also, Wadler [2007] has shown that, given extensionality, induction schemes associated with datatypes can be deduced directly from their Church-encoding. However, to conveniently program with these encodings one needs to use the parametricity conditions in computationally relevant positions. For instance the natural numbers can be

encoded in the Calculus of Constructions as the polymorphic type $\mathbf{N} ::= \forall X : \star. X \to (X \to X) \to X$, since any inhabitant n of this type cannot inspect the parameter X. The free theorem associated with any such Church Numeral $n : \mathbf{N}$ is:

$$\begin{aligned}
&\forall X : \star. \forall P : (X \to \star). \\
&\forall z : X. P\, z \to \\
&\forall s : X \to X. (\forall y : X. P\, (s\, y)) \to \\
&P\, (n\, z\, s)
\end{aligned}$$

which, in extensional theories, can be used to derive the usual induction principle for $n : \mathbb{N}$ [Wadler, 2007]:

$$\forall P : (\mathbb{N} \to \star). (\forall m : \mathbb{N}. P\, (\mathsf{succ}\, m)) \to P\, n$$

Related work

This thesis being an extended version of [Bernardy and Moulin, 2012], it does not reflect the state of the art on Internalized Parametricity. Several related papers have been written since our paper was published in the 2012 LICS Proceedings; we briefly present a few of them below:

Keller and Lasson [2012] extended Relational Parametricity to the Calculus of Inductive Constructions (CIC). They added a new, non-informative, sort hierarchy inhabited by the codomain of parametric relations, which forbids nested application of parametricity. They also prove the Abstraction Theorem for CIC, and provide a Coq tactic for constructing proof terms by parametricity.

Bernardy and Moulin [2013] developed an alternative presentation of the calculus presented in this thesis, in which hypercubes are kept implicit and their dimensions (called *colors*) are *named*; the ability to abstract over dimensions removes the need for some of the technicalities we developed earlier in [Bernardy and Moulin, 2012]. In addition, an *erasure* operator reveals some structural invariants, hence some definitions and proofs may be omitted as they become trivial. In fact, the latter paper was an attempt to make the calculus presented here easier to use as a programming language.

Krishnaswami and Dreyer [2013] built a parametric model of the Extensional Calculus of Constructions. They focus on soundness properties and show how to derive equality results (such as some com-

mon postulates on Church-encoded data) from parametricity conditions. However, the fact that parametricity is modelled in an extensional theory makes it impractical to use their model to build a programming language.

Atkey et al. [2014] describe parametric models of predicative and impredicative Dependent Type Theories in reflexive graphs, which are in turn seen as Categories with Families. In the impredicative case, they how to take advantage of parametricity to derive the existence of initial algebras for all indexed functors.

Outline

After recalling previous work by Bernardy et al. [2010] in chapter 1, we show in chapter 2 how to extend any strong enough Pure Type System $\mathcal{O}$ (such as the Calculus of Constructions) with new rules, including a *Parametricity* Rule, which all have a computational content. More specifically, we describe a new system and show how to adapt the previous result in order to achieve internalization. In the latter subsections, we expose some of the technical problems encountered, and the solutions we found, namely the introduction of hypercubes. In chapter 3 we give a formal presentation of our calculus, and state and prove important meta-properties of our system. In particular, we prove that all instances of the abstraction theorem can be both expressed and proved in the calculus itself. Finally, by defining a reduction-preserving interpretation from our system to the underlying PTS $\mathcal{O}$ in section 3.6, we show how to derive desired meta properties properties such as Church-Rosser and Strong Normalization.

Proofs for free

This chapter is a reminder and synthesis of previous work by Bernardy et al. [2010] and Bernardy and Lasson [2011], which the present work is largely based on.

1.1 Pure type systems

Pure Type Systems (PTSs) are a family of λ-calculi, parameterized by a set of sorts $\mathcal{S}$, a set of axioms $\mathcal{A} \subseteq \mathcal{S} \times \mathcal{S}$ and set of rules $\mathcal{R} \subseteq \mathcal{S} \times \mathcal{S} \times \mathcal{S}$. The various syntactic forms of quantifications (and corresponding abstraction and application) are syntactically unified, and one needs to inspect sorts to identify which form is meant. The axioms $\mathcal{A}$ give the typing rules for sorts, and $\mathcal{R}$ determines which forms of quantification exist in the system. Many systems (*e.g.*, the Calculus of Constructions or System F) are examples of PTSs.

The syntax of PTS terms is the following:

$$
\begin{array}{llcll}
\mathsf{Term} & \ni A, \ldots, U & = & s & \text{sort} \\
& & \mid & x & \text{variable} \\
& & \mid & A\,B & \text{application} \\
& & \mid & \lambda x : A.\, B & \text{abstraction} \\
& & \mid & \forall x : A.\, B & \text{product}
\end{array}
$$

The product $\forall x : A.\, B$ may also be written $A \to B$ when x does not occur free in B. In the rest of this document we assume a given PTS specification $(\mathcal{S}, \mathcal{A}, \mathcal{R})$, and we name the calculus arising from that specification $\mathcal{O}$. In particular, a suitable $\mathcal{O}$ is the Calculus of Constructions, the typing rules of which can be expressed in a PTS fashion by choosing the following specification:

$$
\begin{aligned}
\mathcal{S} &= \{\star, \Box\} \\
\mathcal{A} &= \{(\star, \Box)\} \\
\mathcal{R} &= \{(\star,\star,\star),\ (\star,\Box,\Box),\ (\Box,\star,\star),\ (\Box,\Box,\Box),\ \}
\end{aligned}
$$

$$\frac{}{\vdash s_1 : s_2}\ (s_1, s_2) \in \mathcal{A} \qquad \textsc{Axiom}$$

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B} \qquad \textsc{Weakening}$$

$$\frac{\Gamma \vdash F : (\forall x : A.\, B) \qquad \Gamma \vdash a : A}{\Gamma \vdash F\, a : B[a/x]} \qquad \textsc{Application}$$

$$\frac{\Gamma, x : A \vdash b : B \qquad \Gamma \vdash (\forall x : A.\, B) : s}{\Gamma \vdash (\lambda x : A.\, b) : (\forall x : A.\, B)} \qquad \textsc{Abstraction}$$

$$\frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash (\forall x : A.\, B) : s_3}\ (s_1, s_2, s_3) \in \mathcal{R} \qquad \textsc{Product}$$

$$\frac{\Gamma \vdash A : B \quad \Gamma \vdash B' : s \quad B =_\beta B'}{\Gamma \vdash A : B'} \qquad \textsc{Conversion}$$

$$\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A} \qquad \textsc{Start}$$

Figure 1: Typing rules of the Pure Type System specified by $(\mathcal{S}, \mathcal{A}, \mathcal{R})$

1.2 Logical relations, from PTS to PTS

In this section we recall the relational interpretation of terms and types of the PTS $\mathcal{O}$ into another PTS, here called $[\![\mathcal{O}]\!]$, following the construction of [Bernardy et al., 2010, Bernardy and Lasson, 2011].

In any PTS, types and terms in $\mathcal{O}$ can respectively be interpreted in $[\![\mathcal{O}]\!]$ as predicates and proofs that the terms satisfy the predicates. Each *type* can be interpreted as a predicate that its inhabitants satisfy; and each *term* can be turned into a proof that it satisfies the predicate of its type. Usual presentations of parametricity use binary relations, but for simplicity of notation we present here a unary version. The generalization to arbitrary arity is straightforward, as shown by Bernardy and Lasson [2011].

In the following we define what it means for a term C to satisfy the predicate generated by a type T (which we write $C \in [\![T]\!]$); and the translation from a program C of type T to a proof $[\![C]\!]$ that C satisfies the predicate.

More precisely, we define (Def. 4) two mutually recursive functions $T \mapsto [\![T]\!]$ and $T \mapsto (\cdot \in [\![T]\!])$, by induction on the structure of the raw term T. These interpretations respectively take terms and types in $\mathcal{O}$, and return proofs and propositions in $[\![\mathcal{O}]\!]$. Let $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ be the specifications of the PTS $\mathcal{O}$; then those of $[\![\mathcal{O}]\!]$ are

$$\begin{aligned} \mathcal{S}' &= \mathcal{S} \\ \mathcal{A}' &= \mathcal{A} \\ \mathcal{R}' &= \{(s_1, s_2, s_2) \mid (s_1, s_2, s_3) \in \mathcal{R}\} \cup \mathcal{R} \end{aligned} \tag{2}$$

Before we formally define the interpretation, let us begin by stating the *abstraction* theorem for $\mathcal{O}$ and $[\![\mathcal{O}]\!]$: Any well-typed term of $\mathcal{O}$ is interpreted as a proof that it satisfies the parametricity condition of its type [Bernardy and Lasson, 2011].

Theorem 1 (Abstraction)**.** *If* $\Gamma \vdash_{\mathcal{O}} A : B : s$*, then*

$$[\![\Gamma]\!] \vdash_{[\![\mathcal{O}]\!]} [\![A]\!] : (\{A\} \in [\![B]\!]) : s$$

(Where $\{A\}$ *is* A *in which each free variable* y *in* Γ *is renamed to* y_0 *in the extended context* $[\![\Gamma]\!]$*, defined below.)*

Furthermore, if $\mathcal{O}$ *is consistent, for instance if* $\mathcal{O}$ *is the Calculus of Constructions, then so is* $[\![\mathcal{O}]\!]$ *[Bernardy and Lasson, 2011].*

A property of the translation $[\![\cdot]\!]$ is that whenever $x : A$ is free in T, there are two variables x_0 and x_1 in $[\![T]\!]$, where x_1 witnesses that x_0

satisfies the parametricity condition of its type ($x_1 : x_0 \in [\![A]\!]$). This means that the translation needs to be extended to contexts, as follows:

$$\begin{aligned}[\![\varepsilon]\!] &= \varepsilon \\ [\![\Gamma, x : A]\!] &= [\![\Gamma]\!], x_0 : \{A\}, x_1 : x_0 \in [\![A]\!]\end{aligned}$$

It is important to notice that this definition assumes a *global* renaming from each variable x to fresh variables x_0 and x_1. (The renaming will be made local in further chapters.)

A raw term T in $\mathcal{O}$ is syntactically translated, by mutual induction on its structure, to both a proof term $[\![T]\!]$ in $[\![\mathcal{O}]\!]$, and to a predicate $C \mapsto (C \in [\![T]\!])$. We separate these two interpretations in the presentation below.

- The translation of a variable is done by looking up the corresponding parametric witness in the context.

$$[\![x]\!] = x_1$$

- The case for abstraction adds a witness that the input satisfies the relational interpretation of its type and returns the relational interpretation of the body.

$$[\![\lambda x : A.\, B]\!] = \lambda x_0 : \{A\}.\, \lambda x_1 : x_0 \in [\![A]\!].\, [\![B]\!]$$

- The application follows the same pattern: the function is passed a witness that the argument satisfies the interpretation of its type.

$$[\![A\, B]\!] = [\![A]\!]\, \{B\}\, [\![B]\!]$$

- If the term has another syntactic form, namely a product or a sort, then it is a type (T). Thus we can use λ-abstraction to create a predicate and check that the abstracted variable z satisfies the relational interpretation of the type in the body ($z \in [\![T]\!]$).

$$\begin{aligned}[\![s]\!] &= \lambda z : s.\, z \to s \\ [\![\forall x : A.\, B]\!] &= \lambda z : (\forall x_0 : \{A\}.\, \{B\}). \\ &\qquad \lambda x_0 : \{A\}.\, \lambda x_1 : x_0 \in [\![A]\!]. \\ &\qquad (z\, x) \in [\![B]\!]\end{aligned}$$

We now need to define the proposition $C \in [\![T]\!]$ which, as it can be seen in Thm. 1, is the type of $[\![C]\!]$ for any well-typed $C : T$.

- Because types in a PTS are abstract, no predicate can discriminate between them, hence any predicate over a type C can be used to witness that C satisfies the relational interpretation of its sort s.

$$C \in [\![s]\!] = C \to s$$

- If the type is a product ($\forall x : A.\, B$), then C must be a function, and it satisfies the relational interpretation of its type if and only if it maps satisfying inputs to satisfying outputs.

$$C \in [\![\forall x : A.\, B]\!] = \forall x_0 : \{A\}.\, \forall x_1 : x_0 \in [\![A]\!].\, (C\, x_0) \in [\![B]\!]$$

- For any other syntactic form for a type T, namely a variable, an application or a lambda, $C \in [\![T]\!]$ is defined using the interpretation $[\![\cdot]\!]$ given above: $C \in [\![T]\!] = [\![T]\!]\, \{C\}$.

$$\begin{aligned} C \in [\![x]\!] &= x_1\, \{C\} \\ C \in [\![A\, B]\!] &= [\![A]\!]\, \{B\}\, [\![B]\!]\, \{C\} \\ C \in [\![\lambda x : A.\, B]\!] &= \lambda x_1 : \{C\} \in [\![A]\!].\, [\![B]\!] \end{aligned}$$

A direct reading of Thm. 1 is as a typing judgment about translated terms: if A has type B, then $[\![A]\!]$ has type $\{A\} \in [\![B]\!]$. However, it can also be understood as an abstraction theorem for $\mathcal{O}$: if a program A has type B in Γ, then A satisfies the relational interpretation of its type ($\{A\} \in [\![B]\!]$). Remember that $\{A\}$ is merely the term A, but using variables in $[\![\Gamma]\!]$ instead of Γ. In particular, if A is closed then $\{A\} = A$. If we were to study binary parametricity, $[\![\Gamma]\!]$ would contain two related environments (and witnesses that they are properly related). Therefore A would have two possible interpretations $\{A\}$, each obtained by picking variables out of each copy of the environment, and $[\![A]\!]$ would be a proof that the two possible interpretations of A are related.

One can show by induction on raw terms that whenever $C : T : s$, we have:

$$[\![T]\!] : \{T\} \to s \qquad C \in [\![T]\!] =_\beta [\![T]\!]\, \{C\} : s \qquad \{C\} : \{T\}$$

One may wonder why we mutually define two interpretations, since instead one could be defined from the other using the above equality. The advantage of the distinction, as described by Bernardy and Lasson [2011], is that it makes derivations in $[\![\mathcal{O}]\!]$ follow the same structure

as those in $\mathcal{O}$. Indeed, if we were using the same interpretation both for types and terms, derivations in $[\![\mathcal{O}]\!]$ would be cluttered by extra uses of the conversion rule, as it was earlier presented by Bernardy et al. [2010]. Furthermore, preserving cuts makes the congruence of our model (Lem. 16) trivial.

In general the PTS $[\![\mathcal{O}]\!]$, where parametricity conditions are expressed, extends the source system $\mathcal{O}$. However, for rich enough systems, such as the calculus of constructions, they can be identical [Bernardy et al., 2010, Bernardy and Lasson, 2011]. Indeed, the PTS specifications are then closed under the parametric interpretation, presented at the beginning of this section. We now show how to extend such a system $\mathcal{O}$ to a new calculus $\mathcal{P}$ with internalized parametricity.

Towards internalizing parametricity

In this chapter we describe and motivate our system step by step, starting from a Pure Type System (such as the Calculus of Constructions) and extending it with our new constructions. In this chapter we gradually motivate and informally describe the system we envision. The full specification of our calculus can be found in definitions 3 to 8.

2.1 Aim and example

Let us assume a PTS $\mathcal{Q}$ satisfying equation (2) (*i.e.*, $\mathcal{Q} = [\![\mathcal{Q}]\!]$), such as the Calculus of Constructions. This means that both types and their parametricity conditions can be expressed in $\mathcal{Q}$, one can hope that for every term A of type B, we can get a witness $[\![A]\!]$ that it is parametric ($\{A\} \in [\![B]\!]$). Even though this holds for closed terms, it is not so for open terms, because the context where $[\![A]\!]$ is meaningful is "bigger" than that where A is: for each free variable $x : A$ in Γ, we need a variable $x_1 : x \in [\![A]\!]$ in $[\![\Gamma]\!]$. In other words, given $\Gamma \vdash_{\mathcal{Q}} A : B$ we have $[\![\Gamma]\!] \vdash_{\mathcal{Q}} [\![A]\!] : \{A\} \in [\![B]\!]$. However if we are to use this judgment inside a proof or a program, we are bound to the context encountered, hence we cannot extend it with explicit parametric witnesses for each free variable.

What we really want is to *derive* each free theorem rather than postulating the precise instances, and to be able to rely on parametricity conditions *in the same context*. Therefore, we need the following judgment to be valid:

$$\Gamma \vdash_{\mathcal{Q}} [\![A]\!] : A \in [\![B]\!].$$

The aim of this work is to find a system $\mathcal{P}$ such that the following proposition is verified.

Proposition 1 (Internal Parametricity). *If* $\Gamma \vdash_{\mathcal{P}} A : B$, *then*

$$\Gamma \vdash_{\mathcal{P}} [\![A]\!] : A \in [\![B]\!]$$

That is, the free theorem associated with each inhabited type B can be proved in the system $\mathcal{P}$ *itself, regardless of whether B is closed or not.*

In that case, for any term A, terms of $\mathcal{P}$ can invoke the fact that A is parametric, by writing $[\![A]\!]$. The notations $[\![A]\!]$ and $A \in [\![B]\!]$ for $\mathcal{P}$ will be defined later in this section, following and extending their homonyms in $\mathcal{O}$.

Such a system would allow a full internalization of Reynold's abstraction theorem seen in the introduction, in the sense that variables and implication no longer need to be expressed at the meta-level:

Example 1. *Assume that* $\mathcal{P}$ *extends the Calculus of Constructions. Let us consider the following instance of Internal Parametricity:*

$$\Gamma ::= f : (\forall a : \star.\, a \to a), a : \star, x : a \qquad A ::= f \qquad B ::= \forall a : \star.\, a \to a$$

Then applying internal parametricity gives:

$$\begin{aligned} f : (\forall a : \star.\, a \to a), a : \star, x : a \vdash_{\mathcal{P}} f : \forall a : \star.\, a \to a \quad \Longrightarrow \\ f : (\forall a : \star.\, a \to a), a : \star, x : a \vdash_{\mathcal{P}} [\![f]\!] : \forall a : \star.\, \forall P : a \to \star. \\ \forall x : a.\, P\, x \to \\ P\, (f\, a\, x) \end{aligned}$$

We are thus able to prove that any function of type $\forall a : \star.\, a \to a$ *is an identity, as we hinted at in the introduction. The formulation of the theorem within* $\mathcal{P}$ *and its proof term are as follows.*

$$\begin{aligned} &\mathit{identities} : \forall f : (\forall a : \star.\, a \to a).\, \forall a : \star.\, \forall x : a.\, f\, a\, x \equiv x \\ &\mathit{identities} = \lambda f.\, \lambda a.\, \lambda x.\, [\![f]\!]\, a\, (\cdot \equiv x)\, x\, (\mathit{refl}\, a\, x) \end{aligned}$$

where the infix $\equiv$ *stands for Leibniz equality described in the introduction, and for* $a : \star$ *and* $x : a$, $(\cdot \equiv x)$ *denotes the predicate of terms Leibniz-equal to* x: $(\cdot \equiv x) ::= \forall y : a.\, y \equiv x$; $\mathit{refl}\, a\, x : x \equiv x$ *is merely the identity function* $\lambda P : (\forall x : a.\, \star).\, \lambda p : P\, x.\, p$

If identities *is applied to a "concrete" identity function, such as* $f = \lambda a : \star.\, \lambda x : a.\, x$, *then* $f\, a\, x$ *reduces to* x, *and the theorem specializes to reflexivity of equality:*

$$\mathit{identities}\, f \;:\; \forall a : \star.\, \forall x : a.\, x \equiv x$$

After reduction, the proof no longer mentions $[\![\cdot]\!]$*:*

$$\begin{aligned} \mathit{identities}\, i \quad &\to_\beta \quad \lambda a.\, \lambda x.\, [\![f]\!][\lambda a : \star.\, \lambda x : a.\, x / f]\ a\, (\cdot \equiv x)\, x (\mathit{refl}\, a\, x) \\ &= \quad \lambda a.\, \lambda x.\, [\![\lambda a : \star.\, \lambda x : a.\, x]\!]\ a\, (\cdot \equiv x)\, x (\mathit{refl}\, a\, x) \\ &= \quad \lambda a.\, \lambda x.\, (\lambda a.\, \lambda a_1.\, \lambda x.\, \lambda x_1.\, x_1)\ a\, (\cdot \equiv x)\, x (\mathit{refl}\, a\, x) \\ &\to_\beta \quad \lambda a.\, \lambda x.\, \mathit{refl}\, a\, x \end{aligned}$$

(Where $\to_\beta$ stands for the β reduction in $\mathcal{P}$, which we will define below in Def. *6.)*

It is to be noted that when applying Thm. *1 instead of Internal Parametricity to the above instance, the context Γ is extended to*

$$\begin{array}{ll} f : (\forall A : \star.\, A \to A), & f_1 : \forall A : \star. \forall P : A \to \star. \forall x : A.\, P\, x \to P\, (f\, A\, x), \\ A : \star, & P : A \to \star, \\ x : A, & p : P\, x \end{array}$$

Hence in identities, one cannot rely on a parametricity witness for f without asserting *it. And in a proof assistant, free variables will only ever be instantiated by λ-terms, which are known to be parametric by* Thm. *1.*

However Thm. *1 and internal parametricity coincide on* closed *instances (i.e., when Γ is the empty context).*

2.2 Internalization

We will now give an overview of our system $\mathcal{P}$. This system is obtained by starting from a PTS $\mathcal{O}$ such that $\mathcal{O} = [\![\mathcal{O}]\!]$, for instance the Calculus of Constructions, and adding several constructions. In the present section we give motivations for the new constructions, and present the precise syntax and inference rules of system $\mathcal{P}$ later in definitions 3 to 8. We emphasize that the motivations given in this section are informal, and consistency of the system and other fundamental properties are proved later in sections 3.3 to 3.6.

We have seen that the abstraction theorem (Thm. 1) for PTSs gives us something very close to internal parametricity, except that for each free variable $x : A$ in Γ, we need an explicit witness that x is parametric ($x_1 : x \in [\![A]\!]$) in the environment.

However, we know that every closed term is parametric. Therefore, ultimately, we know that for each possible *concrete* term a that can be substituted for a free variable x, it is possible to construct a concrete term $[\![a]\!]$ to substitute for x_1. This means that the witness of parametricity for x does not need to be given explicitly (if x is bound). Therefore we allow to access such a witness via the new *syntactic* form $[\![x]\!]$. This intuition justifies the addition of the substitution rule

$$[\![x]\!]\,[a/x] = [\![a]\!]$$

as well as the following typing rule, expressing that if x is found in the context, then it is valid to use $[\![x]\!]$, which witnesses that x satisfies the parametricity condition of its type.

$$\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash [\![x]\!] : x \in [\![A]\!]}$$

Note that because of this new construction $[\![\cdot]\!]$, the system $\mathcal{P}$ that we are defining in this section is not a Pure Type System. However it extends any PTS $\mathcal{O}$ such that $\mathcal{O} = [\![\mathcal{O}]\!]$.

At the same time, we must amend the parametric interpretation to keep track of which variables have been assigned an explicit witness, and which variables must wait for a concrete term. For instance in Ex. 1, only the bound variables of the identity f were assigned explicit witnesses. The parametric witnesses of a free variable x is given by our new syntactic construct $[\![x]\!]$, while that of a bound variable y is picked directly from the context as y_1. Hence we need to keep track of free variables when defining the interpretation; we write the list of assignments as an index to $[\![\cdot]\!]$, and extend $A \in [\![B]\!]$ to $A \in [\![B]\!]_\xi$ accordingly. (From here on, we let $[\![A]\!]$ mean $[\![A]\!]_\varnothing$.) For example, abstraction is translated as follows:

$$[\![\lambda x : A.\, B]\!]_\xi = \lambda x_0 : \{A\}_\xi.\, \lambda x_1 : x_0 \in [\![A]\!]_\xi.\, [\![B]\!]_{\xi, x \mapsto (x_0, x_1)}$$

and other cases are modified accordingly. In particular, the interpretation of variables becomes the following[1].

$$\begin{aligned} [\![x]\!]_\xi &= x_1 && \text{if } x \mapsto (x_0, x_1) \in \xi \\ [\![x]\!]_\xi &= [\![x]\!] && \text{if } x \notin \xi \end{aligned}$$

and $\{\cdot\}$ is generalized in a similar fashion: $\{A\}_\xi$ is A where each free variable in $x \in \xi$ is replaced with x_0, while variables that are not in ξ remain untouched.

The difference of treatment between free and bound variables is illustrated in the following example:

$$\begin{aligned} x : A \vdash b : B \quad &\Longrightarrow \quad x : A \vdash [\![b]\!] : b \in [\![B]\!] \\ \vdash \lambda x : A.\, b : \forall x : A.\, B \quad &\Longrightarrow \quad \vdash \lambda x_0 : A.\, \lambda x_1 : x \in [\![A]\!].\, [\![b]\!]_{\{x\}} : \\ &\qquad\qquad \forall x_0 : A.\, \forall x_1 : x \in [\![A]\!].\, b \in [\![B]\!]_{\{x\}} \end{aligned}$$

The above construction solves the issue of context extension. That is, every term A of a PTS $\mathcal{Q}$ can be proved parametric by using $[\![A]\!]$ without extending the context where A is typeable. Another aspect of the result is that, assuming parametricity on variables, the parametricity for all terms can be derived. This means that, in a language featuring parametricity, the parametric construction can be used on any term, but in normal forms, $[\![\cdot]\!]$ only appears on variables, possibly in a nested way.

1 – Careful readers might worry that we discard the index in the second case. An informal justification is that if x has no explicit witness, then the free variables of its type do not either; thus types are preserved by this equation.

Unfortunately, internal parametricity does not quite hold at this stage, after the mere extension of the original calculus with the constructor $\llbracket \cdot \rrbracket$. Indeed, as we show in the next section, Subject Reduction does not hold.

2.3 Parametricity of parametricity

Assuming that $\mathcal{P}$ has Internalized Parametricity (Prop. 1), the fact that all values are parametric is also captured by the following theorem (internalized inside the calculus):

$$\begin{aligned} &\mathit{parametricity} : \forall a : \star. \forall x : a.\, x \in [\![a]\!] \\ &\mathit{parametricity} = \lambda a : \star. \lambda x : a.\, \llbracket x \rrbracket \end{aligned}$$

Since all terms are assumed parametric, it should be possible to apply $[\![\cdot]\!]$ to the above term. For a closed type $A : \star$, consider the term

$$[\![\mathit{parametricity}\ A]\!] = \lambda x_0 : A. \lambda x_1 : x_0 \in [\![A]\!].\, [\![\llbracket x \rrbracket]\!]_{\{x \mapsto (x_0, x_1)\}}$$

So far, we have not defined our meta-operation $[\![\cdot]\!]_\xi$ on the new constructor $\llbracket x \rrbracket$ of our system $\mathcal{P}$ (where x is a free variable). A perhaps natural idea is to exchange the two occurrences of the parametric interpretation, by defining $[\![\llbracket x \rrbracket]\!]_\xi = [\![[\![x]\!]_\xi]\!]$. In our case, that leads to

$$[\![\llbracket x \rrbracket]\!]_{\{x \mapsto (x_0, x_1)\}} = [\![[\![x]\!]_{\{x \mapsto (x_0, x_1)\}}]\!] = \llbracket x_1 \rrbracket$$

which is a proper normal form. Unfortunately, this definition *does not preserve types* (*i.e.*, it breaks Subject Reduction). This can be checked by assuming $x : A$, and by computing the types of the expression before and after reduction. Internal Parametricity gives $\llbracket x \rrbracket : [\![A]\!]\ x$. By Abstraction (giving an explicit parametric witness for x), we get

$$\begin{aligned} [\![\llbracket x \rrbracket]\!]_{\{x \mapsto (x_0, x_1)\}} &: [\![[\![A]\!]\ x]\!]_{\{x \mapsto (x_0, x_1)\}}\ \{[\![x]\!]\}_{\{x \mapsto (x_0, x_1)\}} \\ &: [\![[\![A]\!]]\!]_{\{x \mapsto (x_0, x_1)\}}\ x_0\ [\![x]\!]_{\{x \mapsto (x_0, x_1)\}}\ \llbracket x_0 \rrbracket \\ &: [\![[\![A]\!]]\!]\ x_0\ [\![x]\!]_{\{x \mapsto (x_0, x_1)\}}\ \llbracket x_0 \rrbracket \\ &: [\![[\![A]\!]]\!]\ x_0\ x_1\ \llbracket x_0 \rrbracket \end{aligned} \tag{3}$$

On the other hand, by Abstraction we have $[\![x]\!]_{\{x \mapsto (x_0, x_1)\}} : [\![A]\!]\ x_0$, and by application of Internal Parametricity, we obtain

$$\begin{aligned} [\![[\![x]\!]_{\{x \mapsto (x_0, x_1)\}}]\!] = \llbracket x_1 \rrbracket &: [\![[\![A]\!]\ x_0]\!]\ x_1 \\ &: [\![[\![A]\!]]\!]\ x_0\ \llbracket x_0 \rrbracket\ x_1 \end{aligned} \tag{4}$$

That is, in the above example, the reduction rule suggested above has the effect to swap the second and third arguments to $[\![[\![A]\!]]\!]$ in the type, which means that Subject Reduction would not hold if we were to have the above, naive rule.

However, one observes that, for a *closed* type A, the relation $[\![[\![A]\!]]\!]\ x$ is symmetric: $[\![[\![A]\!]]\!]\ x\ B\ C$ *isomorphic* to $[\![[\![A]\!]]\!]\ x\ C\ B$. Thus the swapping observed above is harmless, and it is sufficient to deal with it in a technical fashion.

Example 2. *For instance, the relation* $[\![[\![(a : \star) \to a \to a]\!]]\!]\ f$ *is symmetric for any* f*. That is,*

$$[\![[\![(a : \star) \to a \to a]\!]]\!]\ f\ f_1\ f_2 \quad \textit{and} \quad [\![[\![(a : \star) \to a \to a]\!]]\!]\ f\ f_2\ f_1$$

are isomorphic for all f_1, f_2 *of type* $[\![(a : \star) \to a \to a]\!]\ f$*.*
Indeed, $[\![[\![(a : \star) \to a \to a]\!]]\!]\ f\ f_1\ f_2$ *expands to*

$$\begin{array}{ll} \forall a : \star. & \forall P : a \to \star. \\ & \forall Q : a \to \star. \forall R : (x : a) \to P\,x \to Q\,x \to \star. \\ \forall x : a. & \forall p : Q\,x. \\ & \forall q : T\,x. \forall r : R\,x\,p\,q. \\ & \quad R\,(f\,a\,x)\,(f_1\,a\,P\,x\,p)\,(f_2\,a\,Q\,x\,q) \end{array}$$

If $\varphi : [\![[\![(a : \star) \to a \to a]\!]]\!]\ f\ f_1\ f_2$*, an inhabitant of*

$$[\![[\![(a : \star) \to a \to a]\!]]\!]\ f\ f_2\ f_1$$

is given by swapping the abstractions of respectively P *and* Q*, and* p *and* q*:*

$$\lambda a.\, \lambda P.\, \lambda Q.\, \lambda R.\, \lambda x.\, \lambda p.\, \lambda q.\, \lambda r.\, \varphi\, a\, Q\, P\, (\lambda x.\, \lambda p.\, \lambda q.\, R\, x\, q\, p)\, x\, q\, p\, r$$

In the light of this observation, we introduce a special-purpose operator (pronounced exchange) $\cdot \ddagger^{\pi}$, which applies the given permutation π to the arguments of relations, and which permutes their types in the same way.

$$\frac{\Gamma \vdash A : B}{\Gamma \vdash A \ddagger^{\pi} : B \ddagger^{\pi}}$$

This rule generalizes the above example to open terms and types. Indeed, when instantiated to Ex. 2, the new operator merely swaps the abstractions:

$$\frac{\vdash \varphi : [\![[\![(a : \star) \to a \to a]\!]]\!]\ f\ f_1\ f_2}{\vdash \varphi \ddagger^{(1,2)} : [\![[\![(a : \star) \to a \to a]\!]]\!]\ f\ f_2\ f_1}$$

Thanks to this operation we can now properly define the parametric interpretation on the constructor $[\![\cdot]\!]$, in a way that preserves types. The

above situation now becomes:

$$[\![[\![x]\!]]\!]_{\{x \mapsto (x_0, x_1)\}} = [\![[\![x]\!]_{\{x \mapsto (x_0, x_1)\}}]\!] \ddagger^{(1,2)}.$$

However, supporting exchange ($\cdot \ddagger$) requires deep changes in the syntax, exposed in the next section.

2.4 A syntax for hypercubes

In order to support the swapping operation, we need to indicate the role of each of the arguments to the relations explicitly, in the syntax. To this end, we amend the abstract syntax, and introduce a new version of application where arguments are tied together in a cubical structure. For instance, the type of $[\![[\![x]\!]]\!]_{\{x \mapsto (x_0, x_1)\}}$, which was written before as $[\![[\![A]\!]]\!]\ x_0\ x_1\ \llbracket x_0 \rrbracket$, is now written

$$[\![[\![A]\!]]\!] \bullet \begin{pmatrix} x_0 & x_1 \\ \llbracket x_0 \rrbracket & \cdot \end{pmatrix}.$$

that is, the 3 arguments of the relation $[\![[\![A]\!]]\!]$ are tied together into an (incomplete) 2×2 matrix. Its counterpart, corresponding to the former $[\![[\![A]\!]]\!]\ x_0\ \llbracket x_0 \rrbracket\ x_1$, can now be obtained by merely transposing the matrix:

$$\left([\![[\![A]\!]]\!] \bullet \begin{pmatrix} x_0 & x_1 \\ \llbracket x_0 \rrbracket & \cdot \end{pmatrix}\right) \ddagger^{(1,2)} =_\beta [\![[\![A]\!]]\!] \bullet \begin{pmatrix} x_0 & \llbracket x_0 \rrbracket \\ x_1 & \cdot \end{pmatrix}$$

One could understand hypercube application as a macro denoting a $(2^n - 1)$-place application. However, we need to make this explicit in the syntax to be able to perform exchanges without extra complication of the analysis of terms. Indeed, having grouped the arguments allows us to massage them all at once in the β-reduction and parametric interpretation; however they should really be read in their "linearized" form, such as the $[\![[\![A]\!]]\!]\ x_0\ x_1\ \llbracket x_0 \rrbracket$ above.

In general, we need to remember the grouping of arguments when applying the relational interpretation. Essentially, one iteration of the relational interpretation transforms an application of an argument into application of two arguments. After a second iteration, there will be four arguments, and 2^n after n iterations. (We must change the abstract syntax of application to group these 2^n arguments together.) Abstraction and product follow the same pattern as application. Hence, we can arrange our bindings as oriented n-cubes in general. Using overbar to

denote cube meta-variables, the syntax becomes the following:

$$\begin{array}{rcll} \mathsf{Term} & = & A\,\bar{B} & \text{application (of hypercubes)} \\ & | & \lambda\bar{x} : \bar{A}.\, B & \text{abstraction (of hypercubes)} \\ & | & \forall\bar{x} : \bar{A}.\, B & \text{function space} \\ & | & \ldots & \end{array}$$

In the above, a binding $\bar{x} : \bar{B}$ introduces 2^n variables x_i, where i is any bit-vector of size n, and n is the dimension of $\bar{B}$. Consider the binding $\bar{x} : \bar{B}$. If $\bar{B}$ has dimension zero, it stands for a single binding $x : B$. If it has dimension 1, it contains a type B_0, and a predicate B_1 over B_0. Abusing matrix notation, one could write

$$\bar{x} : \begin{pmatrix} B_0 \\ B_1 \end{pmatrix} \quad \text{as a shorthand for the two bindings} \quad \begin{pmatrix} x_0 : B_0 \\ x_1 : B_1 x_0 \end{pmatrix}$$

At dimension two, the cube $\bar{B}$ contains a type B_{00}, two predicates B_{01} and B_{10} over B_{00}, and a relation B_{11}, between B_{00}, $B_{10}\,x_{00}$, and $B_{01}\,x_{00}$.

$$\bar{x} : \begin{pmatrix} B_{00} & B_{01} \\ B_{10} & B_{11} \end{pmatrix} \quad \text{means} \quad \begin{pmatrix} x_{00} : B_{00} & x_{01} : B_{01}\,x_{00} \\ x_{10} : B_{10}\,x_{00} & x_{11} : B_{11}\,x_{00}\,x_{01}\,x_{10} \end{pmatrix}$$

Since we refer to each vertex by its position in the hypercube, we define hypercubes of dimension n as mappings from bit-vectors of length n to terms. We write

$$\left[\ i \mapsto B_i\ \right]^n_{i \in \mathbf{2}^n} \qquad \text{and} \qquad \left[\ i \mapsto B_i\ \right]^n_{i \in \mathbf{2}^n - 1}$$

respectively for plain and incomplete cubes (those that lack an element at index 1...1, called top index in the following) of dimension n.

We furthermore need a special syntax for the introduction, elimination and formation of relations, which correspond to application, abstraction and quantification over incomplete cubes. Such a cube is found for example in the type of x_{11} above. Using a check $\check{}$ to denote incomplete cube (*i.e.*, one of those with 2^n-1 vertices) meta-variables:

$$\begin{array}{rcll} \mathsf{Term} & = & A \bullet \check{B} & \text{relation membership} \\ & | & \lambda^\bullet \check{x} : \check{A}.\, B & \text{relation formation} \\ & | & \check{A} \xrightarrow{\bullet} s^n & \text{relation space} \\ & | & \ldots & \end{array}$$

Using this syntax, we can finally write the type of x_{11}, previously linearized as $B_{11}\,x_{00}\,x_{01}\,x_{10}$, in the form we need: $B_{11} \bullet \begin{pmatrix} x_{00} & x_{01} \\ x_{10} & \cdot \end{pmatrix}$. The

type of B_{11} is $\begin{pmatrix} B_{00} & B_{01} \\ B_{10} & \cdot \end{pmatrix} \overset{\bullet}{\to} s$. For a plain cube $\bar{B}$ of arbitrary dimension, we have $x_{1\ldots1} : B_{1\ldots1}\bullet(\bar{x}/\!/1...1)$ and $B_i : (\bar{B}/\!/i) \overset{\bullet}{\to} s$, where $\bar{B}/\!/1...1$ denotes the cube $\bar{B}$ with the top vertex removed. Further generalizing, x_i is a witness that the sub-cube found by removing all the dimensions d such that $i_d = 0$ satisfies the relation B_i:

$$x_i : B_i\bullet(\bar{x}/\!/i)$$

where $\bar{B}/\!/i$ is the cube obtained by discarding the elements of the cube $\bar{B}$ for each dimension d where $i_d = 0$, and then removing the top vertex.

$$\bar{B}/\!/i = \left[\ j \mapsto B_{j\&i}\ \right]_{j\in\mathbf{2}^{\|i\|}-1}^{\|i\|}$$

where $\|i\| = \sum_d i_d$ and & is the pointwise and between bitvectors:

$$\begin{array}{ll} (bj)\&(0i) & = 0(j\&i) \\ (bj)\&(1i) & = b(j\&i) \end{array}$$

B_i is then a relation over the corresponding sub-cube of $\bar{B}$, which is written formally:

$$B_i : (\bar{B}/\!/i) \overset{\bullet}{\to} s$$

Remark. *In this presentation, free theorems, or more generally logical relations, can only take an incomplete cube (of $2^n - 1$ vertices) as argument, whereas their proofs involve applications of full cubes. In particular, partial applications of a parametric relation are not allowed. We should also stress that the syntax is an extension of the underlying PTS, which can be recovered by restricting to cubes of dimension zero.*

2.5 The interpretation of hypercubes

Having given the new syntax of terms, we can express the relational interpretation using this new syntax. The interpretation of a cube increases its dimension; to each element is associated its interpretation:

$$[\![\bar{A}]\!]_{\bar{\zeta}} = \left[\ \begin{array}{l} 0i \mapsto \{A_i\}_{\bar{\zeta}} \\ 1i \mapsto [\![A_i]\!]_{\bar{\zeta}} \end{array}\ \right]_{i\in\mathbf{2}^{\dim\bar{A}}}^{\dim\bar{A}+1}$$

If a binding $\bar{x}$ has been extended by the interpretation, a variable x_i is then interpreted as x_{1i}.

$$[\![x_i]\!]_{\bar{\zeta},x} = x_{1i}$$

The interpretation of terms mentioning full cubes (of size 2^n for some n) is the following:

$$\begin{aligned}
[\![A\ \bar{B}]\!]_{\bar{\xi}} &= [\![A]\!]_{\bar{\xi}}\ [\![\bar{B}]\!]_{\bar{\xi}} \\
[\![\lambda \bar{x} : \bar{A}.\, B]\!]_{\bar{\xi}} &= \lambda \bar{x} : [\![\bar{A}]\!]_{\bar{\xi}}.\, [\![B]\!]_{\bar{\xi}, x \mapsto (x_0, x_1)} \\
C \in [\![\forall \bar{x} : \bar{A}.\, B]\!]_{\bar{\xi}} &= \forall \bar{x} : [\![\bar{A}]\!]_{\bar{\xi}}.\, (C\ (\bar{x}/01...1)) \in [\![B]\!]_{\bar{\xi}, x \mapsto (x_0, x_1)}
\end{aligned}$$

The interpretation of the cubes of size $2^n - 1$ used for relations requires some care. Because the index 1...1 is missing in such a cube, applying the same method as for full cubes leaves two elements missing, at indices 1...1 and 01...1. The former is supposed to be missing (because the resulting cube is also incomplete), but the latter is dependent on the context. Hence we introduce the following notation for interpretation of incomplete cubes where the "missing element" is explicitly specified to be B:

$$([\![\check{A}]\!]_{\bar{\xi}} \oplus B) = \left[\begin{array}{l} 0i \mapsto \{A_i\}_{\bar{\xi}} \\ 1i \mapsto [\![A_i]\!]_{\bar{\xi}} \\ 01...1 \mapsto B \end{array}\right]_{i \in \mathbf{2}^{\dim \check{A} - 1}}^{\dim \check{A} + 1}$$

The parametric interpretation of the special forms for relation formation, membership and product are as follows[2].

$$\begin{aligned}
C \in [\![\check{A} \overset{\bullet}{\to} s]\!]_{\bar{\xi}} &= ([\![\check{A}]\!]_{\bar{\xi}} \oplus C) \overset{\bullet}{\to} s \\
C \in [\![A \bullet \check{B}]\!]_{\bar{\xi}} &= [\![A]\!]_{\bar{\xi}} \bullet ([\![\check{B}]\!]_{\bar{\xi}} \oplus C) \\
[\![\lambda^\bullet \check{x} : \check{A}.\, B]\!]_{\bar{\xi}} &= \lambda^\bullet \check{x} : ([\![\check{A}]\!]_{\bar{\xi}} \oplus (\lambda^\bullet \check{x} : \check{A}.\, B)). \\
&\qquad\qquad x_{01...1} \in [\![B]\!]_{\bar{\xi}, x \mapsto (x_0, x_1)}
\end{aligned}$$

They are a straightforward consequence of the usual parametric interpretation and our choice of grouping arguments in cubes. Readers familiar with realizability interpretations for the Calculus of Constructions (in the style for example of [Paulin-Mohring, 1989]) will notice a similarity here: the interpretation of a function space adds a quantification; and the other forms behave accordingly. Note that the form $A \bullet \check{B}$ is always a type, and therefore we interpret it as such.

2 – Note that the missing element (the right hand-side of the $\oplus$ operator) is always a subterm of the expression we start with.

We now revisit nested parametricity (presented above in section 2.3):

Example 3 (Nested application of $[\![\cdot]\!]$)**.**

$$[\![\textit{parametricity } A]\!] = \lambda \bar{a} : [\![(A)]\!].\, \lceil a_1 \rceil \ddagger^{(01)}$$

where $\bar{a} : [\![(A)]\!]$ *can be understood as* $\begin{pmatrix} a_0 \\ a_1 \end{pmatrix} : \begin{pmatrix} A \\ a_0 \in [\![A]\!] \end{pmatrix}$*. There we have on the one hand*

$$\begin{aligned} [\![\textit{parametricity } A]\!] : (\textit{parametricity } A) &\in [\![\forall a : A.\ \in [\![A]\!]]\!] \\ &= (\lambda a : A.\, \lceil a \rceil) \in [\![\forall a : A.\ \in [\![A]\!]]\!] \\ &= \forall \bar{a} : [\![(A)]\!].\, [\![A]\!]^2_\bullet \begin{pmatrix} a_0 & a_1 \\ \lceil a_0 \rceil & \cdot \end{pmatrix} \end{aligned}$$

while on the other hand

$$a_1 : a_0 \in [\![A]\!] = [\![A]\!]_\bullet \begin{pmatrix} a_0 \\ \cdot \end{pmatrix}$$

hence

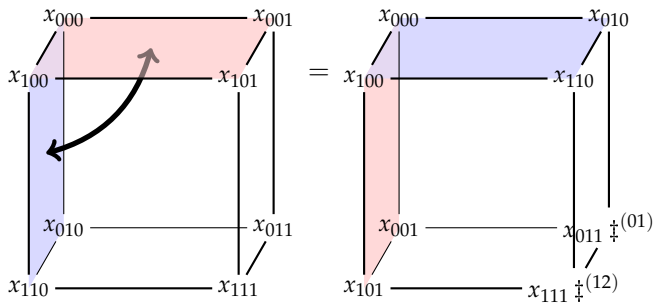
$$\begin{aligned} \lceil a_1 \rceil \ddagger^{(01)} : (a_1 \in [\![a_0 \in [\![A]\!]]\!]) \ddagger^{(01)} &= [\![A]\!]^2_\bullet \begin{pmatrix} a_0 & \lceil a_0 \rceil \\ a_1 & \cdot \end{pmatrix} \ddagger^{(01)} \\ &= [\![A]\!]^2_\bullet \begin{pmatrix} a_0 & a_1 \\ \lceil a_0 \rceil & \cdot \end{pmatrix} \end{aligned}$$

So Subject Reduction no longer fails as it did for the system without hypercubes presented in section 2.3.

2.6 Exchanging dimensions

Given the above definition of cubes, we can take advantage of the fact that vertices are tied to the structure and define an operation that applies an arbitrary permutation of its dimensions. For dimension $n = 0$ or $n = 1$, there is no non-trivial permutation. In the case of a square ($n = 2$), there is only one permutation, which is a simple swapping of the elements at indices 01 and 10. For higher dimensions ($n \geq 3$), the elements of the cube are multidimensional themselves (the dimension of an element at index i is $\|i\|$). Thus, one must take care to perform the exchange properly for each element. For instance, performing an

exchange of dimensions 1 and 2 in a cube $\bar{x}$ for $n = 3$ involves exchanging dimensions 0 and 1 of the element x_{011}. Indeed, exchanging the dimensions 1 and 2 in the cube has the effect of exchanging dimensions in the square occupied by x_{011}; so an exchange has to be performed on x_{011} to restore the cube structure. Geometrically, exchanging the dimensions as above corresponds to twisting the cube: two faces are swapped, and the two other are twisted. The situation is shown graphically in the following picture.



In general, applying a permutation π on the dimensions of a cube $\bar{C}$ is done as follows:

Definition 1 (Cube exchange)**.**

$$\bar{C}\ddagger^{\pi} = \left[\ i \mapsto C_{\pi(i)}\ddagger^{\pi/i}\ \right]_{i\in\mathbf{2}^{\dim\bar{C}}}^{\dim\bar{C}}$$

Where π/i stands for the permutation π restricted to the dimensions d where $i_d = 1$.
Incomplete cubes are permuted in the same way (simply omitting the top vertex).

Definition 2. *If π is a permutation $\{d \mapsto x_d\}$,*
$\pi/i = \text{canon}\{d \mapsto x_d \mid i_d = 1\}$, where canon *maps the domain and codomain of the function $\{d \mapsto x_d \mid i_d = 1\}$ to the set $\{0..\|i\| - 1\}$, preserving the order. Renaming the dimensions in the permutation ensures that sub-cubes can be treated just like normal cubes.*

Example 4. *If $\pi = \{0 \mapsto 0, 1 \mapsto 2, 2 \mapsto 1\}$ swaps dimensions 1 and 2, we have*

i	$\{d \mapsto \pi(d) \mid i_d = 1\}$	π/i
001	$\{2 \mapsto 1\}$	$\{0 \mapsto 0\}$
010	$\{1 \mapsto 2\}$	$\{0 \mapsto 0\}$
100	$\{0 \mapsto 0\}$	$\{0 \mapsto 0\}$
011	$\{1 \mapsto 2, 2 \mapsto 1\}$	$\{0 \mapsto 1, 1 \mapsto 0\}$
101	$\{0 \mapsto 0, 2 \mapsto 1\}$	$\{0 \mapsto 0, 1 \mapsto 1\}$
110	$\{0 \mapsto 0, 1 \mapsto 2\}$	$\{0 \mapsto 0, 1 \mapsto 1\}$

Applying a permutation to terms is then a matter of permuting all the cubes encountered:

$$\begin{aligned}(A\,\bar{B})\,\ddagger^{\pi}_{\xi} &= A\,\ddagger^{\pi}_{\xi}\ \bar{B}\,\ddagger^{\pi}_{\xi}\\ (\lambda\bar{x} : \bar{A}.\,B)\,\ddagger^{\pi}_{\xi} &= \lambda\bar{x} : \bar{A}\,\ddagger^{\pi}_{\xi}.\,B[\bar{x}\,\ddagger^{\pi}/\bar{x}]\,\ddagger^{\pi}_{\xi,x}\\ (\forall\bar{x} : \bar{A}.\,B)\,\ddagger^{\pi}_{\xi} &= \forall\bar{x} : \bar{A}\,\ddagger^{\pi}_{\xi}.\,B[\bar{x}\,\ddagger^{\pi}/\bar{x}]\,\ddagger^{\pi}_{\xi,x}\end{aligned}$$

(and similarly for the incomplete cubes). It remains to explain the interaction with the special constructs, $[\![\cdot]\!]$ and $\cdot\,\ddagger^{\cdot}$ itself. We do so by listing four laws which hold in our calculus.

The first law is not surprising: the composition of exchanges is the exchange of the composition.

$$A\,\ddagger^{\rho}\,\ddagger^{\pi} =_{\beta} A\,\ddagger^{\pi\circ\rho} \tag{5}$$

Regarding the interactions between $[\![\cdot]\!]$ and $\cdot\,\ddagger^{\pi}$, recall first that the relational interpretation adds one dimension to cubes. By convention, the dimension added by $[\![\cdot]\!]$ is at index 0, and all other dimensions are shifted by one. Therefore, the relational interpretation of an exchange merely lifts the exchange out, and shifts indices by one in its permutation, leaving dimension 0 intact.

$$[\![A\,\ddagger^{\pi}]\!] =_{\beta} [\![A]\!]\,\ddagger^{\pi+1} \tag{6}$$

where $\pi+1$ denotes the permutation $\{d \mapsto \pi(d-1) \mid 0 < d \leq \dim\pi\}$.

The law that motivates the introduction of exchanges is the following:

$$[\![[\![A]\!]_{\xi}]\!] =_{\beta} [\![[\![A]\!]_{\xi}]\!]\,\ddagger^{(01)} \tag{7}$$

This law can also be explained by the convention that $[\![\cdot]\!]$ always increases each existing dimension and inserts a new dimension 0. By commuting the uses of parametricity, dimensions are swapped, and the exchange operator restores the order.

Last, one can also simplify exchanges in the presence of symmetric terms. We know that a term $[\![A]\!]^n$ is symmetric in its n first dimensions. Thus, applying a permutation that touches only dimensions $0..n-1$ to such a term has no effect. Formally, we have:

$$[\![A]\!]^n\,\ddagger^{(x_1\ x_2\ldots x_m)} =_{\beta} [\![A]\!]^n \qquad \text{if } \forall i \in 1..m, x_i < n \tag{8}$$

We have argued before that it suffices to provide parametricity only for variables, and that the construct $[\![\cdot]\!]$ acts as a "macro" on other constructs. The situation is the same in the presence of dimension exchanges: equation (6) explains how to compute the parametricity witness of an exchange. For the $\cdot\,\ddagger^{\pi}$ construct, the situation is analogous:

it suffices to provide the construct for variables, possibly enclosed by $[\![\cdot]\!]$ themselves, while it is a macro on all other forms.

The reason is that the above laws give a way to compute the exchange for any term which is not a parametricity witness (the result is given in Def. 5). When we want to be explicit about exchange being the syntactic construct, we write simply $\mathbf{x}\dagger^{\pi}$. The syntax fragment for parametricity and exchanges is as follows.

$$
\begin{array}{llcll}
\mathsf{Var} & \in x, y, z & & & \\
\mathsf{Param} & \in \mathbf{x} & ::= & x & \text{variable} \\
 & & \mid & [\![\mathbf{x}]\!] & \text{parametric witness} \\
\mathsf{Term} & \in a, \ldots, u & ::= & \mathbf{x}\dagger^{\pi} & \text{permutation of dimensions} \\
 & & \mid & \ldots &
\end{array}
$$

2.7 Dimension checks

If a permutation acts on dimensions 0 to $n-1$, every cube where it is applied to *must* exhibit at least n dimensions. So far we have not discussed this restriction, which is the final feature of the system to present. To implement it we choose to amend the syntax and annotate sorts with the dimension of the type which inhabits it. Since the sort s at dimension n is written s^n, we can capture the restriction in the following exchange rule.

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash B : s^n}{\Gamma \vdash A \ddagger^{\pi} : B \ddagger^{\pi}} \dim(\pi) \leq n$$

(However, as with the PARAM rule, only the version where the term A is a *variable* is added to our typing rules; this is enough, since the general rule can be derived.)

If a type inhabits a sort of dimension n, all the quantifications found inside the type must be over cubes of dimension at least n. This is enforced by modifying the product rule as follows:

$$\frac{\Gamma \vdash \bar{A} : s_1^n \qquad \Gamma, \bar{x} : \bar{A} \vdash B : s_2^m}{\Gamma \vdash (\forall \bar{x} : \bar{A}.\, B) : s_3^{m \sqcap n}}$$
$$\textsc{Product}\ (s_1, s_2, s_3) \in \mathcal{R}$$

Similarly, relations found in the type must be over cubes of dimension n.

A calculus with an internal parametricity theorem

Having concluded the informal presentation of our system $\mathcal{P}$, we focus now on a detailed description and will end with proofs of some fundamental meta-properties such as Confluence (Thm. 3), Strong Normalization (Thm. 9), and Consistency (Thm. 8).

3.1 Definitions

We start this section with the full definition of system $\mathcal{P}$, parameterized on a PTS specification $(\mathcal{S}, \mathcal{A}, \mathcal{R})$.

Definition 3 (Abstract syntax of $\mathcal{P}$)**.**

Sort	$\ni s, s_1, s_2, s_3$	$::=$	$\mathcal{S}$	
Var	$\in x, y, z$			
Param	$\in \mathbf{x}$	$::=$	x	*variable*
		$\mid$	$[\![\mathbf{x}]\!]$	*parametric witness*
Term	$\in a, \ldots, u$	$::=$	$\mathbf{x}\,\dagger^{\pi}$	*permutation of dimensions*
	$A, \ldots, U$	$\mid$	s^n	*sort at dimension n*
		$\mid$	$A\,\bar{B}$	*application (of hypercubes)*
		$\mid$	$\lambda \bar{x} : \bar{A}.\, B$	*abstraction (of hypercubes)*
		$\mid$	$\forall \bar{x} : \bar{A}.\, B$	*function space*
		$\mid$	$A \bullet \check{B}$	*relation membership*
		$\mid$	$\lambda^{\bullet} \check{x} : \check{A}.\, B$	*relation formation*
		$\mid$	$\check{A} \xrightarrow{\bullet} s^n$	*relation space*
Cube	$\ni \bar{A}$	$::=$	$\left[\ i \mapsto A_i\ \right]^n_{i \in \mathbf{2}^n}$	*cube of size* 2^n
Cube′	$\ni \check{A}$	$::=$	$\left[\ i \mapsto A_i\ \right]^n_{i \in \mathbf{2}^n - 1}$	*cube of size* $2^n - 1$
Context	$\ni \Gamma, \Delta$	$::=$	ε	*empty context*
		$\mid$	$\Gamma, x : A$	*context extension*

Where $\left[\ i \mapsto A_i\ \right]^n_{i \in \mathbf{2}^n}$ *(resp.* $\left[\ i \mapsto A_i\ \right]^n_{i \in \mathbf{2}^n - 1}$*) denote a balanced binary tree (resp. a balanced binary tree without the lower-right leaf) of depth n,*

where for each bit-vector i of length n, the vertex A_i is the leaf reached from the root by following the left child on 1's and right one on 0's.

The cube bindings can be defined formally once we introduce some convenient notations:

- $\mathbf{2}^n$ *stands for all bit-vectors of size* n*; and* $\mathbf{2}^n - 1$ *stands for all bit-vectors of size* n*, except* 1...1*.*
- $\mathrm{ind}(\bar{A})$ *stands for* $\mathbf{2}^{\dim \bar{A}}$*; and* $\mathrm{ind}(\check{A})$ *stands for* $\mathbf{2}^{\dim \check{A}} - 1$*.*
- $\bar{x} : \bar{A}$ *stands for the bindings* $x_i : A_i \bullet (\bar{x} /\!\!/ i)$ *where* $i \in \mathrm{ind}(\bar{A})$*; and* $\check{x} : \check{A}$ *stands for the bindings* $x_i : A_i \bullet (\check{x} /\!\!/ i)$ *where* $i \in \mathrm{ind}(\check{A})$*.*
- *Similarly,* $\bar{A} : s^n$ *stands for* $A_i : \bar{A} /\!\!/ i \xrightarrow{\bullet} s^{\|i\|}$ *and* $\check{A} : s^n$ *stands for* $A_i : \check{A} /\!\!/ i \xrightarrow{\bullet} s^{\|i\|}$*.*

Definition 4 (Relational interpretation of raw terms)**.**

$$
\begin{aligned}
[\![[\![x]\!]^n]\!]_\xi &= [\![x]\!]^{n+1} \qquad \textit{(in particular, } [\![x]\!]_\xi = [\![x]\!] \textit{ for } n = 0) && \textit{if } x \notin \xi \\
[\![[\![x_i]\!]^n]\!]_\xi &= [\![x_{1i}]\!]^n \dagger^{(0..n)} \textit{ (in particular, } [\![x_i]\!]_\xi = x_{1i} \textit{ for } n = 0) && \textit{if } x \in \xi \\
[\![\mathbf{x} \dagger^\pi]\!]_\xi &= [\![\mathbf{x}]\!]_\xi \dagger^{\pi+1} \\
[\![\lambda \bar{x} : \bar{A}.\, B]\!]_\xi &= \lambda \bar{x} : [\![\bar{A}]\!]_\xi . [\![B]\!]_{\xi, x \mapsto (x_0, x_1)} \\
[\![\lambda^\bullet \check{x} : \check{A}.\, B]\!]_\xi &= \lambda^\bullet \check{x} : ([\![\check{A}]\!]_\xi \oplus (\lambda^\bullet \check{x} : \check{A}.\, B)). \\
&\qquad\qquad x_{01\ldots1} \in [\![B]\!]_{\xi, x \mapsto (x_0, x_1)} \\
[\![A\ \bar{B}]\!]_\xi &= [\![A]\!]_\xi\ [\![\bar{B}]\!]_\xi \\
[\![T]\!]_\xi &= \lambda \check{z} : \begin{pmatrix} T \\ \cdot \end{pmatrix} . z_0 \in [\![T]\!]_\xi && \textit{if } T \textit{ is } \forall, \bullet \textit{ or } s^n
\end{aligned}
$$

$$
\begin{aligned}
C \in [\![s^n]\!]_\xi &= \begin{pmatrix} C \\ \cdot \end{pmatrix} \xrightarrow{\bullet} s^{n+1} \\
C \in [\![\forall \bar{x} : \bar{A}.\, B]\!]_\xi &= \forall \bar{x} : [\![\bar{A}]\!]_\xi . (C\,(\bar{x}/01\ldots1)) \in [\![B]\!]_{\xi, x \mapsto (x_0, x_1)} \\
C \in [\![\check{A} \xrightarrow{\bullet} s^n]\!]_\xi &= ([\![\check{A}]\!]_\xi \oplus C) \xrightarrow{\bullet} s^{n+1} \\
C \in [\![A \bullet \check{B}]\!]_\xi &= [\![A]\!]_\xi \bullet ([\![\check{B}]\!]_\xi \oplus C) \\
C \in [\![T]\!]_\xi &= [\![T]\!]_\xi \bullet \begin{pmatrix} C \\ \cdot \end{pmatrix} && \textit{if } T \textit{ is not } \forall, \bullet \textit{ nor } s^n
\end{aligned}
$$

$$
\begin{aligned}
[\![\varepsilon]\!]_\xi &= \varepsilon \\
[\![\Gamma, x : A]\!]_{\xi, x \mapsto (x_0, x_1)} &= [\![\Gamma]\!]_\xi, x_0 : A, x_1 : x_0 \in [\![A]\!]_\xi && \textit{if } x \in \xi \\
[\![\Gamma, x : A]\!]_\xi &= [\![\Gamma]\!]_\xi, x : A && \textit{if } x \notin \xi
\end{aligned}
$$

$$[\![\bar{A}]\!]_\xi = \left[\begin{array}{l} 0i \mapsto \{A_i\}_\xi \\ 1i \mapsto [\![A_i]\!]_\xi \end{array}\right]^{\dim \bar{A}+1}_{i \in \mathbf{2}^{\dim \bar{A}}}$$

$$([\![\check{A}]\!]_\xi \oplus B) = \left[\begin{array}{l} 0i \mapsto \{A_i\}_\xi \\ 1i \mapsto [\![A_i]\!]_\xi \\ 01...1 \mapsto B \end{array}\right]^{\dim \check{A}+1}_{i \in \mathbf{2}^{\dim \check{A}}-1}$$

Definition 5 (Term exchange)**.**

$$\begin{aligned}
[\![x]\!]^n \dagger^\rho \ddagger^\pi_\xi &= [\![x]\!]^n \dagger^\rho && \text{if } x \in \xi \\
[\![x]\!]^n \dagger^\rho \ddagger^\pi_\xi &= [\![x]\!]^n \dagger^{\mathrm{normal}_n(\pi \circ \rho)} && \text{if } x \notin \xi \\
(A\ \bar{B}) \ddagger^\pi_\xi &= A \ddagger^\pi_\xi\ \bar{B} \ddagger^\pi_\xi \\
(\lambda \bar{x} : \bar{A}.\, B) \ddagger^\pi_\xi &= \lambda \bar{x} : \bar{A} \ddagger^\pi_\xi . B[\bar{x} \ddagger^\pi / \bar{x}] \ddagger^\pi_{\xi,x} \\
(\forall \bar{x} : \bar{A}.\, B) \ddagger^\pi_\xi &= \forall \bar{x} : \bar{A} \ddagger^\pi_\xi . B[\bar{x} \ddagger^\pi / \bar{x}] \ddagger^\pi_{\xi,x} \\
(A \bullet \bar{B}) \ddagger^\pi_\xi &= A \ddagger^\pi_\xi \bullet \bar{B} \ddagger^\pi_\xi \\
(\lambda^\bullet \check{x} : \check{A}.\, B) \ddagger^\pi_\xi &= \lambda^\bullet \check{x} : \check{A} \ddagger^\pi_\xi . B[\bar{x} \ddagger^\pi / \bar{x}] \ddagger^\pi_{\xi,x} \\
(\check{A} \xrightarrow{\bullet} s^n) \ddagger^\pi_\xi &= \check{A} \ddagger^\pi_\xi \xrightarrow{\bullet} s^n \\
s^n \ddagger^\pi_\xi &= s^n
\end{aligned}$$

Where $\mathrm{normal}_n(\pi)$ *removes all cycles of* π *entirely contained in* $0..n-1$*.*

The β-reduction of the underlying PTS extends naturally to hypercube redexes.

Definition 6 (Reduction)**.**

$$\begin{aligned}
(\lambda \bar{x} : \bar{A}.\, b)\, \bar{a} &\longrightarrow b[\bar{a}/\bar{x}] \\
(\lambda^\bullet \check{x} : \check{A}.\, b) \bullet \check{a} &\longrightarrow b[\check{a}/\check{x}]
\end{aligned}$$

Where $b[\bar{a}/\bar{x}]$ *(resp.* $b[\check{a}/\check{x}]$*) denotes the* $2^{\dim \bar{a}}$ *(resp.* $2^{\dim \check{a}} - 1$*) substitutions* $b[x_i/a_i \mid i \in \mathbf{2}^{\dim \bar{a}}]$ *(resp.* $b[x_i/a_i \mid i \in \mathbf{2}^{\dim \check{a}} - 1]$*).*

We do not specify a reduction strategy, and the β*-reduction* $\longrightarrow$ *can be applied anywhere in a term, including under abstraction or application.*

We write $=_\beta$ *the reflexive, symmetric, transitive closure of the reduction* $\longrightarrow$*.*

Definition 7 (Substitution)**.** *In addition to the usual congruence rules, we extend the substitution meta-operation to our two new syntactic constructs.*

$$\begin{aligned}
[\![x]\!]\,[a/x] &= [\![a]\!]_\varnothing \\
x \dagger^\pi [a/x] &= a \ddagger^\pi_\varnothing
\end{aligned}$$

Definition 8 (Typing rules of $\mathcal{P}$)**.**

$$\frac{}{\vdash s_1^n : s_2^n}\ (s_1, s_2) \in \mathcal{A} \qquad \textsc{Axiom}$$

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash C : s^n}{\Gamma, x : C \vdash A : B} \qquad \textsc{Weakening}$$

$$\frac{\Gamma \vdash F : (\check{A} \overset{\bullet}{\to} s^n) \qquad \Gamma \vdash \check{a} : \check{A}}{\Gamma \vdash F \bullet \check{a} : s^n} \qquad \textsc{Rel-elim}$$

$$\frac{\Gamma, \check{x} : \check{A} \vdash B : s^n \qquad \Gamma \vdash \check{A} : s^n}{\Gamma \vdash (\lambda^\bullet \check{x} : \check{A}.\, B) : (\check{A} \overset{\bullet}{\to} s^n)} \qquad \textsc{Rel-intro}$$

$$\frac{\Gamma \vdash \check{A} : s_1^n}{\Gamma \vdash (\check{A} \overset{\bullet}{\to} s_1^n) : s_2^n}\ (s_1, s_2) \in \mathcal{A} \qquad \textsc{Rel-form}$$

$$\frac{\Gamma \vdash F : (\forall \bar{x} : \bar{A}.\, B) \qquad \Gamma \vdash \bar{a} : \bar{A}}{\Gamma \vdash F\, \bar{a} : B[\bar{a}/\bar{x}]} \qquad \textsc{Application}$$

$$\frac{\Gamma, \bar{x} : \bar{A} \vdash b : B \qquad \Gamma \vdash (\forall \bar{x} : \bar{A}.\, B) : s^n}{\Gamma \vdash (\lambda \bar{x} : \bar{A}.\, b) : (\forall \bar{x} : \bar{A}.\, B)} \qquad \textsc{Abstraction}$$

$$\frac{\Gamma \vdash \bar{A} : s_1^n \qquad \Gamma, \bar{x} : \bar{A} \vdash B : s_2^m}{\Gamma \vdash (\forall \bar{x} : \bar{A}.\, B) : s_3^{m \sqcap n}}\ (s_1, s_2, s_3) \in \mathcal{R} \qquad \textsc{Product}$$

$$\frac{\Gamma \vdash A : B \qquad \Gamma \vdash B' : s^n \qquad B =_\beta B'}{\Gamma \vdash A : B'} \qquad \textsc{Conversion}$$

$$\frac{\Gamma \vdash A : s^n}{\Gamma, x : A \vdash x : A} \qquad \textsc{Start}$$

$$\frac{\Gamma \vdash \mathbf{x} : A}{\Gamma \vdash [\![\mathbf{x}]\!] : \mathbf{x} \in [\![A]\!]_\varnothing} \qquad \textsc{Param}$$

$$\frac{\Gamma \vdash \mathbf{x} : A \qquad \Gamma \vdash A : s^n}{\Gamma \vdash \mathbf{x} \dagger^\pi : A \ddagger^\pi}\ \dim(\pi) \leq n \qquad \textsc{Exchange}$$

Where the typing judgment $\Gamma \vdash \bar{a} : \bar{A}$ *stands for the conjunction of the judgments* $\Gamma \vdash a_i : A_i \bullet (\bar{a} /\!\!/ i)$ *for* $i \in \mathrm{ind}(\bar{A})$*; and* $\Gamma \in \check{a} : \check{A}$ *stands for the conjunction of the judgments* $\Gamma \vdash a_i : A_i \bullet (\check{a} /\!\!/ i)$ *for* $i \in \mathrm{ind}(\check{A})$*.*

The syntactic changes made to the system require results to be adapted accordingly. In the case of Ex. 1, (proving that any function of type $\forall a : \star.\, a \to a$ is an identity), the definition of Equality must be amended to make it inhabit $\star^1$. This mostly involves augmenting the dimension of cubes by adding unit types as indices:

$$Eq : \forall a : \star.\, a \to \begin{pmatrix} a \\ \cdot \end{pmatrix} \overset{\bullet}{\to} \star^1$$

$$Eq = \lambda a : \star.\, \lambda x : A.\, \lambda^\bullet \begin{pmatrix} y \\ \cdot \end{pmatrix} : \begin{pmatrix} a \\ \cdot \end{pmatrix}.\, \forall \begin{pmatrix} - \\ P \end{pmatrix} : \left(\begin{pmatrix} a \\ \cdot \end{pmatrix}^{\top} \overset{\bullet}{\to} \star^1 \right).$$
$$\left(P \bullet \begin{pmatrix} x \\ \cdot \end{pmatrix}^{\top} \right) \to P \bullet \begin{pmatrix} y \\ \cdot \end{pmatrix}$$

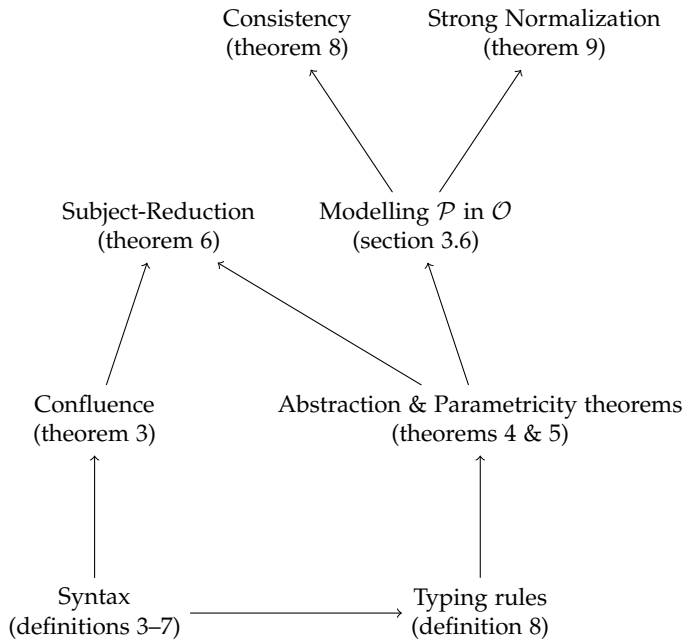
The proof term needs fewer amendments:

$$\begin{aligned} &\mathit{identities} : \forall f : (\forall a : \star . a \to a). \\ &\qquad\qquad \forall a : \star . \forall x : a . Eq\, a\, (f\, a\, x) \bullet \begin{pmatrix} x \\ \cdot \end{pmatrix} \\ &\mathit{identities} = \lambda f . \lambda a . \lambda x . . [\![f]\!] \begin{pmatrix} a \\ Eq\, a\, x \end{pmatrix} \begin{pmatrix} x \\ refl\, a\, x \end{pmatrix} \end{aligned}$$

$$[\![f]\!] : \forall \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} : \begin{pmatrix} \star \\ \bullet \\ a_0 \to \star^1 \end{pmatrix} . \forall \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} : \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} . a_1 \bullet (f a_0 x_0)$$

We have now defined our system $\mathcal{P}$. In the remainder of this chapter we prove the main meta-theoretic results about the system. More precisely, we prove Confluence in section 3.3, the Abstraction and Parametricity theorems in section 3.4, and Subject Reduction in section 3.5. We then define in section 3.6 a reduction-preserving interpretation of $\mathcal{P}$ into the underlying PTS $\mathcal{O}$, hence model the former in the latter. This model is done by introducing explicit witnesses of parametricity for all variables. Provided that Consistency and Strong Normalization hold for $\mathcal{O}$ (for instance when $\mathcal{O}$ is the Calculus of Constructions), we can then derive from the model that they also hold for our system $\mathcal{P}$.

Dependencies between these results can be summarized by the following directed graph:



3.2 Properties of the Parametric Interpretation

We start by proving weakening and commutation lemmas for our parametric interpretation. These lemmas are used to prove Confluence (section 3.3), and Abstraction and Parametricity theorems (section 3.4).

Lemma 1. *For each term A and each variable z not free in A, we have:*

i) $\llbracket A \rrbracket_{\xi, z \mapsto (z_0, z_1)} = \llbracket A \rrbracket_{\xi}$, *and*

ii) $\{a\}_{\xi, z \mapsto (z_0, z_1)} \in \llbracket A \rrbracket_{\xi, z \mapsto (z_0, z_1)} = \{a\}_{\xi} \in \llbracket A \rrbracket_{\xi}$ *for all terms a.*

Proof. By simultaneous induction on the structure of the raw term A. Details can be found in the appendix. □

$\cdot \dagger^{\pi}$ commutes with $\llbracket \cdot \rrbracket$, but the permutation needs to be lifted.

Lemma 2. *For each term $A : s^m$ and each ρ of dimension at most m, we have:*

$$\llbracket A \ddagger_{\zeta}^{\rho} \rrbracket_{\xi} = \llbracket A \rrbracket_{\xi} \ddagger_{\zeta}^{1+\rho}$$

Proof. By induction on the structure of the raw term A. Details can be found in the appendix. □

When exchanging two occurrences of the parametric interpretation, one needs to permute the cube variables that are explicit in both interpretations:

Lemma 3. *For each term A, we have:*

$$[\![[\![A]\!]_{\bar{\xi}}]\!]_{\zeta} = [\![[\![A]\!]_{\zeta}]\!]_{\bar{\xi}}[\bar{x}\dagger^{(01)}/\bar{x} \mid x \in \xi \cap \zeta]\,\ddagger^{(01)}_{\xi\cap\zeta}$$

Proof. By structural induction on the raw term A. Details can be found in the appendix. □

In particular, when ξ is empty:

Corollary 1. *For each term A, we have:*

$$[\![[\![A]\!]^m]\!]_{\zeta} = [\![[\![A]\!]_{\zeta}]\!]^m\,\ddagger^{(0\ldots m)}$$

Note that equation (7) *is a special case of this result, taking $m = 0$.*

The parametric interpretation commutes with the substitution, but a special treatment is required when the variable to be substituted for is either free or known to the interpretation.

Lemma 4 ($[\![\cdot]\!]$ and substitution, part 1)**.** *For each term A, and each variable z not in ξ, we have:*

i) $\quad [\![A[u/z_i]]\!]_{\xi} = [\![A]\!]_{\xi,z}[\{u\}_{\xi}/z_{0i}][[\![u]\!]_{\xi}/z_{1i}]$, *and*

ii) $\quad \{a[u/z_i]\}_{\xi} \in [\![A[u/z_i]]\!]_{\xi} = (\{a\}_{\xi,z} \in [\![A]\!]_{\xi,z})[\{u\}_{\xi}/z_{0i}][[\![u]\!]_{\xi}/z_{1i}]$.

Lemma 5 ($[\![\cdot]\!]$ and substitution, part 2)**.** *For each term A, for variable z not free in A or contained in ξ, we have:*

i) $\quad [\![A[u/z_i]]\!]_{\xi} = [\![A]\!]_{\xi}[\{u\}_{\xi}/z_{0i}]$, *and*

ii) $\quad \{a[u/z_i]\}_{\xi} \in [\![A[u/z_i]]\!]_{\xi} = (\{a\}_{\xi} \in [\![A]\!]_{\xi})[\{u\}_{\xi}/z_{0i}]$.

Proof. By simultaneous induction on the structure of the raw term A. Details can be found in the appendix. □

The last lemma of this section states that our parametric interpretation *uniformly* expands cubes:

Lemma 6 (Symmetry)**.** *For each term A, $[\![A]\!]^n$ is symmetric in its n first dimensions. More specifically,*

i) $\quad [\![A]\!]^n_\xi \ddagger^\pi_\xi = [\![A]\!]^n_\xi \ddagger^{\mathrm{normal}_n(\pi)}_\xi$, *and*

ii) $\quad (a \in [\![A]\!]^n_\xi) \ddagger^\pi_\xi = (a \in [\![A]\!]^n_\xi) \ddagger^{\mathrm{normal}_n(\pi)}_\xi$

Proof. By simultaneous induction on the structure of the raw term A. Details can be found in the appendix. □

Lemma 7 ($\cdot \ddagger^\cdot$ and substitution)**.** *If ξ does not contain either z or any of the free variables of E, then*

$$A[E/z] \ddagger^\pi_\xi = A \ddagger^\pi_\xi [E/z] \text{ for all } \pi.$$

Proof. By induction on A. The only interesting case is the one for variables, with $x = z$:

$$\begin{aligned} [\![z]\!]^n \dagger^\rho [E/z] \ddagger^\pi_\xi &= [\![E]\!]^n \ddagger^\rho \ddagger^\pi_\xi \\ &= [\![E]\!]^n \ddagger^{\mathrm{normal}_n(\pi \circ \rho)} \quad \text{by Lem. 6} \\ &= [\![z]\!]^n \ddagger^{\mathrm{normal}_n(\pi \circ \rho)} [E/z] \\ &= [\![z]\!]^n \dagger^\rho \ddagger^\pi_\xi [E/z] \end{aligned}$$

□

Lemma 8 (Substitution)**.**

$$A[E/z][E'/z'] = A[E'/z'][E[E'/z']/z]$$

Proof. By induction on A; the only non-trivial case is for the parametric witnesses $[\![z]\!]^n$:

$$\begin{aligned} [\![z]\!]^n \dagger^\pi [E/z][E'/z'] &= [\![E]\!]^n_\varnothing [E'/z'] \ddagger^\pi \\ &= [\![E[E'/z']]\!]^n_\varnothing \ddagger^\pi = [\![z]\!]^n \dagger^\pi [E'/z'][E[E'/z']/z] \end{aligned}$$

by lemmas 5 and 7. □

3.3 Confluence

We now check that the Church-Rosser property holds, that is, we verify that the order in which the reductions are performed does not matter. To prove this property, we define a *parallel reduction* (following the Tait/Martin-Löf technique), and show that the diamond property holds for this reduction.

Definition 9 (Parallel nested reduction)**.**

$$\textsc{Refl}\ \frac{}{A \triangleright A}$$

$$\beta\ \frac{b \triangleright b' \qquad \bar{a} \triangleright \bar{a}'}{(\lambda \bar{x} : \bar{A}.\, b)\, \bar{a} \triangleright b'[\bar{a}'/\bar{x}]} \qquad \beta^{\bullet}\ \frac{b \triangleright b' \qquad \check{a} \triangleright \check{a}'}{(\lambda^{\bullet} \check{x} : \check{A}.\, b) \bullet \check{a} \triangleright b'[\check{a}'/\check{x}]}$$

$$\textsc{App-Cong}\ \frac{F \triangleright F' \qquad \bar{a} \triangleright \bar{a}'}{F\, \bar{a} \triangleright F'\, \bar{a}'} \qquad \textsc{App}^{\bullet}\textsc{-Cong}\ \frac{F \triangleright F' \qquad \check{a} \triangleright \check{a}'}{F \bullet \check{a} \triangleright F' \bullet \check{a}'}$$

$$\textsc{Abs-Cong}\ \frac{\bar{A} \triangleright \bar{A}' \qquad b \triangleright b'}{\lambda \bar{x} : \bar{A}.\, b \triangleright \lambda \bar{x} : \bar{A}'.\, b'} \qquad \textsc{Abs}^{\bullet}\textsc{-Cong}\ \frac{\check{A} \triangleright \check{A}' \qquad b \triangleright b'}{\lambda^{\bullet} \check{x} : \check{A}.\, b \triangleright \lambda^{\bullet} \check{x} : \check{A}'.\, b'}$$

$$\textsc{All-Cong}\ \frac{\bar{A} \triangleright \bar{A}' \qquad B \triangleright B'}{\forall \bar{x} : \bar{A}.\, B \triangleright \forall \bar{x} : \bar{A}'.\, B'} \qquad \textsc{All}^{\bullet}\textsc{-Cong}\ \frac{\check{A} \triangleright \check{A}'}{\check{A} \xrightarrow{\bullet} s^n \triangleright \check{A}' \xrightarrow{\bullet} s^n}$$

With $\bar{A} \triangleright \bar{A}'$ iff. for all i, $A_i \triangleright A'_i$ (and similarly for $\check{A} \triangleright \check{A}'$).

We now need to prove congruence lemmas for the parallel reduction $\triangleright$, for each of our 3 meta-operators: parametric interpretation $[\![\cdot]\!]$, term exchange $\cdot\ddagger$, and substitution.

Lemma 9 (Congruence of $[\![\cdot]\!]$)**.** *If $A \triangleright A'$, then for all ξ,*

i) $[\![A]\!]_{\xi} \triangleright [\![A']\!]_{\xi}$, *and*

ii) $a \in [\![A]\!]_{\xi} \triangleright a' \in [\![A']\!]_{\xi}$ *for all $a \triangleright a'$*

Proof. By induction on $A \triangleright A'$:

- The case of Refl is trivial.
- For β, one expects

  $$[\![(\lambda \bar{x} : \bar{A}.\, b)\, \bar{a}]\!]_{\xi} \triangleright [\![b'[\bar{a}'/\bar{x}]]\!]_{\xi},$$

  knowing $b \triangleright b'$ and $\bar{a} \triangleright \bar{a}'$.

$$\begin{array}{ll} & [\![(\lambda \bar{x} : \bar{A}.\, b)\, \bar{a}]\!]_{\xi} \\ = & \quad \{\text{by def. of } [\![\cdot]\!]_{\xi}\} \\ & (\lambda \bar{x} : [\![\bar{A}]\!]_{\xi}.\, [\![b]\!]_{\xi,x})\, [\![\bar{a}]\!]_{\xi} \\ \triangleright & \quad \{\text{by } \beta, \textsc{Refl} \text{ and IH}\} \\ & [\![b']\!]_{\xi,x}[[\![\bar{a}']\!]_{\xi}/\bar{x}] \\ = & \quad \{\text{by Lem. 4}\} \\ & [\![b'[\bar{a}'/\bar{x}]]\!]_{\xi} \end{array}$$

- The case of $\beta^\bullet$ is similar.
- The cases of $\star$-Cong are straightforward using the definition of $[\![\cdot]\!]$. □

Lemma 10 (Congruence of $\cdot \ddagger^{\cdot}$)**.** *If $A \triangleright A'$, then for all ξ and π, one has*

$$A \ddagger_\xi^\pi \triangleright A' \ddagger_\xi^\pi$$

Proof. By induction on $A \triangleright A'$. The only interesting cases are for the β and $\beta^\bullet$-reductions. For β ($\beta^\bullet$ is similar), we have

$$\begin{array}{ll} & ((\lambda \bar{x} : \bar{A}.\, b)\, \bar{a}) \ddagger_\xi^\pi \\ = & \quad \{\text{by def. of } \cdot \ddagger_\xi^\pi\} \\ & (\lambda \bar{x} : \bar{A} \ddagger_\xi^\pi .\, b[\bar{x} \ddagger_\xi^\pi / \bar{x}] \ddagger_{\xi,x}^\pi)\, \bar{a} \ddagger_\xi^\pi \\ \triangleright & \quad \{\text{by } \beta, \textsc{Refl} \text{ and IH}\} \\ & b'[\bar{x} \ddagger_\xi^\pi / \bar{x}] \ddagger_{\xi,x}^\pi [\bar{a}' \ddagger_\xi^\pi / \bar{x}] \\ = & \quad b' \ddagger_{\xi,x}^\pi [\bar{a}' \ddagger_\xi^\pi / \bar{x} \ddagger_\xi^\pi] \\ = & \quad b'[\bar{a}'/\bar{x}] \ddagger_\xi^\pi \end{array}$$

□

Lemma 11 (Congruence of substitution)**.** *If $A \triangleright A'$ and $E \triangleright E'$, then*

$$A[E/z] \triangleright A'[E'/z].$$

Proof. By induction on $A \triangleright A'$:

- For Refl, the expected result follows from an induction on A (using n times Lem. 9 and Lem. 10 for the case $[\![z]\!]^n \ddagger^\pi$).
- For β, one expects

  $$((\lambda \bar{x} : \bar{A}.\, b)\, \bar{a})[E/z] \triangleright b'[\bar{a}'/\bar{x}][E/z],$$

  knowing $b \triangleright b'$ and $\bar{a} \triangleright \bar{a}'$. We have

  $$\begin{array}{ll} & ((\lambda \bar{x} : \bar{A}.\, b)\, \bar{a})[E/z] \\ = & \quad \{\text{by def. of the substitution}\} \\ & (\lambda \bar{x} : A[E/z].\, b[E/z])\, \bar{a}[E/z] \\ \triangleright & \quad \{\text{by } \beta \text{ and IH}\} \\ & b'[E'/z][\bar{a}'[E'/z]/\bar{x}] \\ = & \quad \{\text{by Lem. 8}\} \\ & b'[\bar{a}'/\bar{x}][E'/z] \end{array}$$

- The case of $\beta^\bullet$ is similar.

- The cases of $\star$-Cong stem from straightforward uses of induction hypotheses. □

Theorem 2 (Diamond property)**.** *The rewriting system ($\triangleright$) has the diamond property. In other words, for each A, B, B' such that $B \triangleleft A \triangleright B'$, there exists C such that $B \triangleright C \triangleleft B'$*

Proof. By induction on the derivations:

- If one of the derivations ends with Refl, one has either $A = B$, or $A = B'$. We pick $C = B'$ in the former case and $C = B$ in the latter.

- If one of the derivations ends with Abs-Cong, All-Cong, Abs•-Cong or All•-Cong, the other one has to end with the same rule, and the result is a straightforward use of the induction hypothesis.

- If one of the derivations ends with App-Cong, the other one has to end with App-Cong, or with β. The first case is straightforward; in the second one, one has

$$(\lambda \bar{x} : \bar{A}'.\, b')\, \bar{a}' \;\triangleleft\; (\lambda \bar{x} : \bar{A}.\, b)\, \bar{a} \;\triangleright\; b''[\bar{a}''/\bar{x}]$$

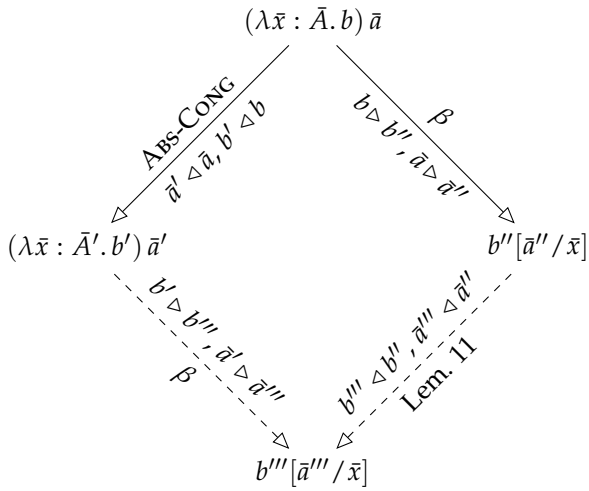
$$\text{with } \lambda \bar{x} : \bar{A}'.\, b' \;\triangleleft\; \lambda \bar{x} : \bar{A}.\, b, \quad b \triangleright b'' \quad \text{and} \quad \bar{a}' \triangleleft \bar{a} \triangleright \bar{a}''$$

  The situation is summarized in the diagram below. In more details, the end of the derivation of $\lambda \bar{x} : \bar{A}'.\, b' \;\triangleleft\; \lambda \bar{x} : \bar{A}.\, b$ has to be either Abs-Cong, or Refl. In the first case (the last one is similar), one has $\bar{A}' \triangleleft \bar{A}$ and $b' \triangleleft b$.

  By induction hypothesis there exist b''', $\bar{a}'''$ such that $b' \triangleright b''' \triangleleft b''$ and $\bar{a}' \triangleright \bar{a}''' \triangleleft \bar{a}''$.

  The result follows by β and Lem. 11:

$$(\lambda \bar{x} : \bar{A}'.\, b')\, \bar{a}' \;\triangleright\; b'''[\bar{a}'''/\bar{x}] \;\triangleleft\; b''[\bar{a}''/\bar{x}]$$



- The case for App•-Cong is similar.
- If both derivations end with the same β or $\beta^\bullet$ rule, the result is a straightforward use of the induction hypothesis and Lem. 11. □

Theorem 3 (Church-Rosser property)**.** *Our calculus system has the confluence (Church-Rosser) property that is, for each A, B, B' such that $B \longleftarrow^\star A \longrightarrow^\star B'$, there exists C such that $B \longrightarrow^\star C \longleftarrow^\star B'$*

Proof. Direct consequence of Thm. 2, using the equality $\triangleright^\star = \longrightarrow^\star$. □

3.4 Abstraction

In this section we check that our main goal, the integration of parametricity (see Prop. 1), is achieved by the design that we propose. (In particular, internalized parametricity holds for the Param rule itself.) At the same time, we check that the abstraction theorem also holds for our calculus. We do so by proving Lem. 12, which subsumes both theorems.

Lemma 12 (Generalized abstraction)**.** *Assuming that ξ conforms to Γ,*

i) $\Gamma \vdash A : B \;\Rightarrow\; [\![\Gamma]\!]_\xi \vdash [\![A]\!]_\xi : \{A\}_\xi \in [\![B]\!]_\xi$

ii) $\Gamma \vdash A : B \;\Rightarrow\; [\![\Gamma]\!]_\xi \vdash \{A\}_\xi : \{B\}_\xi$

iii) $\Gamma \vdash B : s^n \;\Rightarrow\; [\![\Gamma]\!]_\xi, x : B \vdash x \in [\![B]\!]_\xi : s^{n+1}$

Proof. The proof is done by simultaneous induction on the derivation tree, and is similar to the proof of the Abstraction Theorem by Bernardy

and Lasson [2011]. The new parts occur in the special handling of the START and PARAM rules. The proof of each sub-lemma can be sketched as follows (the full proof can be found in the appendix):

i) The cases of abstraction and application stem from the fact that their respective relational interpretations follow the same pattern as the relational interpretation of the product. The case of a variable x (START) is more tricky: if $x \in \xi$, then the context contains an explicit witness of parametricity for x. This witness is used to justify the translated judgment. If $x \notin \xi$, then we can use the parametricity rule on x to translate the typing judgment. The PARAM rule is handled similarly, with the additional complexity that an exchange of dimensions must be added when $x \notin \xi$.

ii) This sub-lemma is used to justify weakening of contexts in the other sub-lemmas. It is a consequence of the thinning lemma and the fact that the interpretation of types in always well-typed (see the third item below).

iii) This sub-lemma expresses that if B is a well-sorted type, then so is $x \in [\![B]\!]$. It is easy to convince oneself of that result by checking that the translation of a type always yields a relation, and that the translation of a relation is itself a relation. □

Remark. *In summary, and roughly speaking,* Lem. *12 replaces the occurrences of* START *(resp.* PARAM*) for variables not in ξ by* PARAM *(resp. nested* PARAM + EXCHANGE *). Occurrences on* START *(resp.* PARAM*) for variables in ξ are preserved.*

Theorem 4 (Abstraction)**.**

i) $\Gamma \vdash A : B \;\Rightarrow\; [\![\Gamma]\!]_\xi \vdash [\![A]\!]_\xi : (\{A\}_\xi \in [\![B]\!]_\xi)$*, where ξ contains all the variables in* Γ*.*

ii) *Furthermore, if the original judgment makes no use of* PARAM*, the resulting judgment does not either.*

Proof.

i) Direct consequence of Lem. 12i.

ii) In the proof of Lem. 12i, if ξ is full, then the target derivation trees contains PARAM iff. PARAM occurs in the derivation tree for $\Gamma \vdash A : B$. □

Theorem 5 (Parametricity)**.** *Each term, no matter if is closed or opened, satisfies the parametricity condition of its type:*

$$\Gamma \vdash A : B \;\Rightarrow\; \Gamma \vdash [\![A]\!] : (A \in [\![B]\!])$$

Proof. Take ξ empty in Lem. 12i. (We recall that $[\![\Gamma]\!]_\varnothing = \Gamma$.) □

Definition 10. *ξ conforms to Γ iff. ξ contains a suffix of Γ.*

Remark. *$[\![\cdot]\!]$ preserves conforming indices: if ξ conforms to Γ and A is well-typed in Γ, the definition of $[\![A]\!]_\xi$ makes only recursive calls with conforming substitutions.*

Proof. By induction on the typing derivation. In the definition of $[\![\cdot]\!]_\xi$, every bound variable in a term is added to the index ξ in recursive calls. □

3.5 Subject Reduction

In this section we prove Subject Reduction (preservation of types). Since parametricity acts as a typing rule for $[\![\cdot]\!]$, Subject Reduction for our calculus stems directly from it. We start by discussing basic properties generally attributed to PTSs, on which Subject Reduction (Thm. 6) depends on.

The weakening of contexts behaves in our calculus exactly in the same way as in all PTSs. Indeed, the usual thinning lemma holds.

Lemma 13 (Thinning)**.** *Let Γ and Δ be legal contexts such that $\Gamma \subseteq \Delta$. Then $\Gamma \vdash A : B \Longrightarrow \Delta \vdash A : B$.*

Proof. As in [Barendregt, 1992, Lem. 5.2.12]. □

The generation lemma for our calculus must account for the new parametricity construct.

Lemma 14 (Generation)**.** *The statement of the lemma is the same as that of the generation lemma for PTS [Barendregt, 1992, Lem. 5.2.13], but with the additional case for the* Param *rule:*

- *If $\Gamma \vdash [\![x]\!] : C$ then there exists B such that $\Gamma \vdash B : s^n$, $(x : B) \in \Gamma$, and $C =_\beta x \in [\![B]\!]$.*

Proof. As in [Barendregt, 1992]:

- We follow the derivation $\Gamma \vdash [\![x]\!] : C$ until $[\![x]\!]$ is introduced. It can only be done by the following rule

$$\frac{\Delta \vdash B : s_n}{\Delta, x : B \vdash [\![x]\!] : x \in [\![B]\!]}\ \textsc{Param}$$

  with $C =_\beta x \in [\![B]\!]$, and $(\Delta, \bar{x} : B) \subseteq \Gamma$. The conclusion stems from Lem. 13. □

Theorem 6 (Subject Reduction). *If $A \longrightarrow A'$ and $\Gamma \vdash A : T$, then*

$$\Gamma \vdash A' : T$$

Proof. Most of the technicalities of the proof by Barendregt [1992], concern β-reduction, and are not changed by our addition of parametricity.

Hence we discuss here only the handling of the parametricity construct: our task is to check that substitution a concrete term a for x in $[\![x]\!]$ preserves the type of the expression.

Facing a term such as $[\![x]\!]$ in context Γ, we know by generation that it must have type $x \in [\![B]\!]$ (for some type B valid in Γ, and $x : B$). We can then prove that substituting a term a of type B' (where B' is convertible to B) for x preserves the type of the expression. Indeed, the expression then reduces to $[\![a]\!]$, which has type $a \in [\![B']\!]$ by Thm. 5. In turn, $a \in [\![B']\!]$ is convertible to $x \in [\![B]\!]$ by Lem. 9. □

3.6 Reduction-preserving model into the underlying PTS

In this section we present a formalization of the intuitive model presented in section 2.2. We developed a "high-level" calculus $\mathcal{P}$ suitable to internalize parametricity results; we now model our system $\mathcal{P}$ into the underlying PTS $\mathcal{O}$, which can be seen as "low-level" in that context.

Each term is mapped to a term where parametricity witnesses are passed explicitly. Simultaneously, contexts are extended with explicit witnesses: in a first approximation, each binding $x : A$ is replaced by a multiple binding $x : A, \check{x} : x \in [\![A]\!]$. This means that $[\![x]\!]$ can be interpreted by the corresponding variable $\check{x}$ in the context. In fact, this is really what the term $[\![x]\!]$ means, as shown by the reduction rule $[\![x]\!]\,[u/x] \longrightarrow [\![u]\!]$.

The following table shows how some example terms can be interpreted (for the sake of readability we omit type annotations in the abstractions, since they play no role in these examples):

original term A	its interpretation $\langle\!\vert A \vert\!\rangle$
$\lambda x.\, [\![x]\!]$	$\lambda x.\, \lambda \check{x}.\, \check{x}$
$(\lambda x.\, [\![x]\!])(yz)$	$(\lambda x.\, \lambda \check{x}.\, \check{x})(yz)(\check{y}z\check{z})$
$(\lambda x.\, [\![x]\!])(\lambda y.\, [\![y]\!])$	$(\lambda x.\, \lambda \check{x}.\, \check{x})\ (\lambda y.\, \lambda \check{y}.\, \check{y})$
	$(\lambda y.\, \lambda \check{y}.\, \lambda \check{y}'.\, \lambda \check{y}'_2.\, \check{y}'_2)$

Note that the third row in the above table shows how an instance of nested parametricity is modelled: we add explicit witnesses of level two.

Given that the interpretation is sound with respect to $\mathcal{O}$ (Thm. 7) and that it preserves reductions (Lem. 16), we obtain Strong Normalization (Thm. 9). The rest of the section is devoted to defining the model formally, and proving its soundness.

In general, the transformation is not trivial, because of the interaction between functions and their arguments, occurring in the App rule. If a function uses parametricity on one of its argument, calls to the function must also compute explicit parametricity witnesses. (This may in turn trigger the need for more explicit witnesses at the call site). Further, if the function is passed to another function, this will create further needs for explicit witnesses.

As we have seen above, each binding $x : A$ should be replaced by $x : A, \ldots, \check{x}^n : x \in [\![A]\!]^n$ for some n. Our main task is to compute an n that would be big enough to make all the parametricity witnesses $[\![x]\!]^k$ explicit. To do so, we use an intermediate representation of the typing derivation, containing some *constraints* on the n's, by annotation of the derivation tree, as in figure 2. We assume without loss of generality that variable names are distinct, so the n's are given by a (partial) valuation $\epsilon : \mathsf{Var} \to \mathbb{N}$ defined on each *cube* variable. This annotation of the derivation with constraints is an instance of a technique known as type-based analysis [Svenningsson, 2007].

$$\frac{\Gamma \vdash F : (\forall \bar{x} : \bar{A}.\, B) \qquad \mathfrak{t} :: \Gamma \vdash \bar{a} : \bar{A} \qquad \{e + \epsilon(x) \leq \epsilon(y) \mid e \leq \epsilon(y) \in \mathfrak{t}\}}{\Gamma \vdash F\,\bar{a} : B[\bar{a}/\bar{x}]}$$
App

$$\frac{\Gamma \vdash F : (\forall^\bullet \check{x} : \check{A}.\, s^n) \qquad \mathfrak{t} :: \Gamma \vdash \check{a} : \check{A} \qquad \{e + \epsilon(x) \leq \epsilon(y) \mid e \leq \epsilon(y) \in \mathfrak{t}\}}{\Gamma \vdash F \bullet \check{a} : s^n}$$
Rel-Elim

$$\frac{\Gamma \vdash A : s^m \qquad n \leq \epsilon(x)}{\Gamma, x : A \vdash [\![x]\!]^n \ddagger^\pi : (x \in [\![A]\!]^n) \ddagger^\pi} \dim \pi \leq m + n$$
Param/n

Figure 2: Typing rules extended with constraints on the valuation ϵ. Rules omitted here remain unchanged (see Def. 8). The notation $e \leq \epsilon(y) \in \mathfrak{t}$ expresses that the constraint appears in the sub-derivation $\mathfrak{t}$. (For the sake of conciseness, we merged the rules Start, Param and Exchange into Param/n here.)

The App and Rel-Elim rules require special care. Indeed, we need to "lift" the inequalities of the right sub-tree $\mathfrak{t}$, since if F has to be extended to a term of type $\forall [\![x : A]\!]^n.\, B$, then it has to be fed with

n extra parametricity witnesses $[\![a]\!] \cdots [\![a]\!]^n$, hence the context has to be extended enough to contain $\check{y}^n$, for each y free in a. Note that the constraints $e + \epsilon(x) \leq \epsilon(y)$ we add in the Application and Rel-Elim rule are more restrictive than the corresponding $e \leq \epsilon(y)$ that are in $\mathfrak{t}$, so one can simply ignore the latter.

We need to check that the system of constraints has a solution. In fact, the simplex it defines is unbounded: indeed, the only place where a variable appears on the left-hand side of a constraint is in Application and Rel-Elim when we "lift by x" the constraints in the sub-tree $\mathfrak{t}$; It cannot create any cycle, since x does not appear in $\mathfrak{t}$.

With our notion of cubes instead of usual bindings, extending the context with an explicit witness corresponds to adding one dimension to the cube. However, we *a priori* only need to access one of the new vertices, the one that has the new dimension set to one. Hence in general, each of the $2^{\dim \bar{A}}$ vertices x_i of a binding $\bar{x} : \bar{A}$ will be extended with $x_{ji} : x_i \in [\![A_i \bullet (\bar{x} /\!\!/ i)]\!]^k$ for $0 \leq k \leq \epsilon(x)$ and $j = 0^{\epsilon(x)-k}1^k$.

Permutations on variables yield yet another difficulty, as one can see in the example $\lambda x : A.\, \lambda y_1 : x \in [\![A]\!].\, \llbracket y_1 \rrbracket \ddagger^{(12)}$. (The cubes have been flattened for the sake of readability.) Here, $\llbracket y_1 \rrbracket \ddagger^{(12)} : [\![A]\!]^2\, x\, y_1\, \llbracket x \rrbracket$ while $\llbracket y_1 \rrbracket : [\![A]\!]^2\, x\, \llbracket x \rrbracket\, y_1$. Our solution is to not only extend the context with explicit parametricity witnesses, but also with explicit *permuted* parametricity witnesses. Hence a possible interpretation of the previous term in the naked system $\mathcal{O}$ is the following:

$$\begin{array}{l} \lambda x_0 : A. \lambda x_1 : ([\![A]\!]\, x_0). \\ \qquad \lambda y_{01} : ([\![A]\!]\, x_0). \lambda y_{11} : ([\![A]\!]^2\, x_0\, x_1\, y_{01}). \\ \qquad\qquad \lambda y_{11}^{(12)} : ([\![A]\!]^2\, x_0\, y_{01}\, x_1). \quad y_{11}^{(12)} \end{array}$$

We are not focusing on the minimal extension here, and we add witnesses for each possible permutation. It is however possible to refine this extension, since for instance the relations are symmetric in the new dimensions, hence we can ignore permutation cycles that are entirely contained in these new dimensions.

Definition 11 (Interpretation which inserts explicit witnesses)**.** *Writing $\mathfrak{S}_n$ to be the group of permutations on $\{0,\dots,n-1\}$,*

$$
\begin{aligned}
\langle\!|s^n|\!\rangle &= s \\
\langle\!|\, [\![x_i]\!]^n \dagger^\pi \,|\!\rangle &= x^\pi_{ji} \quad \textit{where } j = \overbrace{0\dots0\underbrace{1\dots1}_{n}}^{\epsilon(x)} \\
\langle\!|\lambda\bar{x} : \bar{A}.\, B|\!\rangle &= \lambda\langle\!|\bar{x} : \bar{A}|\!\rangle.\, \langle\!|B|\!\rangle \\
\langle\!|\forall\bar{x} : \bar{A}.\, B|\!\rangle &= \forall\langle\!|\bar{x} : \bar{A}|\!\rangle.\, \langle\!|B|\!\rangle \\
\langle\!|F\ ^x\bar{a}|\!\rangle &= \langle\!|F|\!\rangle\ \{\langle\!|\, [\![a_i]\!]^k \ddagger^\pi \,|\!\rangle \mid i \in \mathrm{ind}(\bar{a}), k \le \epsilon(x), \\
&\qquad\qquad \pi \in \mathfrak{S}_{k+\dim\bar{a}} \ \} \\
\langle\!|\lambda^\bullet\check{x} : \check{A}.\, B|\!\rangle &= \lambda\langle\!|\check{x} : \check{A}|\!\rangle.\, \langle\!|B|\!\rangle \\
\langle\!|\forall^\bullet\check{x} : \check{A}.\, s^n|\!\rangle &= \forall\langle\!|\check{x} : \check{A}|\!\rangle.\, s \\
\langle\!|F\bullet^x\check{A}|\!\rangle &= \langle\!|F|\!\rangle\ \{\langle\!|\, [\![a_i]\!]^k \ddagger^\pi \,|\!\rangle \mid i \in \mathrm{ind}(\check{a}), k \le \epsilon(x), \\
&\qquad\qquad \pi \in \mathfrak{S}_{k+\dim\check{a}} \ \} \\
\hline
\langle\!|-|\!\rangle &= - \\
\langle\!|\Gamma, x_i : A|\!\rangle &= \langle\!|\Gamma|\!\rangle, \langle\!|x_i : A|\!\rangle
\end{aligned}
$$

We introduce a new macro $\langle\!|x_i : A|\!\rangle$, which expands to the following multiple bindings:

$$
\begin{aligned}
\langle\!|x_i : A : s^n|\!\rangle = \{x^\pi_{ji} : \langle\!|(x_i \in [\![A]\!]^k) \ddagger^\pi \,|\!\rangle \mid\ & k \le \epsilon(x), \\
& j = 0^{\epsilon(x)-k}1^k, \\
& \pi \in \mathfrak{S}_{k+n} \ \}
\end{aligned}
$$

Bindings of cube variables are merely "flattened", using our previously defined macro:

$$
\begin{aligned}
\langle\!|\bar{x} : \bar{A}|\!\rangle &= \{\langle\!|x_i : A_i\bullet(\bar{x}/\!\!/i)|\!\rangle \mid i \in \mathrm{ind}(\bar{A})\} \\
\langle\!|\check{x} : \check{A}|\!\rangle &= \{\langle\!|x_i : A_i\bullet(\check{x}/\!\!/i)|\!\rangle \mid i \in \mathrm{ind}(\check{A})\}
\end{aligned}
$$

The essence of the model defined by $\langle\!|\cdot|\!\rangle$ is that a parametricity witness $[\![x_i]\!]^n \dagger^\pi$ is adequately modeled by the variable $x^\pi_{0\dots01\dots1i}$, that is, if x has type A, then $x_1 : x \in [\![A]\!]$, etc.

Lemma 15 ($\langle\!|\cdot|\!\rangle$ and substitution)**.**

$$\langle\!|A[a_i/x_i]|\!\rangle = \langle\!|A|\!\rangle\,[\langle\!|\, [\![a_i]\!]^k \ddagger^\pi \,|\!\rangle / x^\pi_{ji}, k \le \epsilon(x), j = 0^{\epsilon(x)-k}1^k, \pi \in \dots]$$

Proof. By induction on A; we illustrate how the proof proceeds by showing only the case for variables, since all the other cases stem from straightforward uses of the induction hypotheses.

$$\begin{aligned}\langle\!| \llbracket x_i \rrbracket^n \dagger^\pi [a_i / x_i] |\!\rangle &= \langle\!| \llbracket a_i \rrbracket^n \ddagger^\pi |\!\rangle \\ &= x_{ji}^\pi [\langle\!| \llbracket a_i \rrbracket^n \ddagger^\pi |\!\rangle / x_{ji}^\pi] \\ &= \langle\!| \llbracket x_i \rrbracket^n \dagger^\pi |\!\rangle [\langle\!| \llbracket a_i \rrbracket^k \ddagger^\pi |\!\rangle / x_{ji}^\pi, \dots]\end{aligned}$$

□

Lemma 16 (Congruence of $\langle\!| \cdot |\!\rangle$)**.** *If $\Gamma \vdash A : B$ with $A \longrightarrow A'$, then*

$$\langle\!| A |\!\rangle \longrightarrow^+ \langle\!| A' |\!\rangle.$$

Proof. By induction on $A \longrightarrow A'$. □

Finally, we are now able to prove the soundness of our model, by proving that the transformation yields well-typed terms in $\mathcal{O}$.

Theorem 7 (Soundness)**.** *If $\Gamma \vdash_{\mathcal{P}} A : B$, then*

$$\langle\!| \Gamma |\!\rangle \vdash_{\mathcal{O}} \langle\!| A |\!\rangle : \langle\!| B |\!\rangle.$$

Proof. We proceed by induction on the derivation; however the proof requires a stronger induction hypothesis when the derivation $\Gamma \vdash_{\mathcal{P}} A : B$ starts with the Application rule. See the appendix for details. □

Theorem 8 (Consistency)**.** *If $\mathcal{O}$ is consistent, then so is $\mathcal{P}$, our system extended with* Param*.*

Proof. Since $\langle\!| \cdot |\!\rangle$ transports the empty type from $\mathcal{P}$ to $\mathcal{O}$, by Thm. 7 any inhabitant of the empty type in $\mathcal{P}$ would give one in $\mathcal{O}$. □

Theorem 9 (Strong Normalization)**.** *If $\mathcal{O}$ is strongly normalizing, then so is $\mathcal{P}$.*

Proof. Assume $\Gamma \vdash A : B$ and consider a chain of reductions $A \longrightarrow^n A'$. We have $\langle\!| A |\!\rangle \longrightarrow^m \langle\!| A' |\!\rangle$, and $m \geq n$ by Lem. 16. We also have that $\langle\!| A |\!\rangle$ is typeable in $\mathcal{O}$, by Thm. 7. Therefore, only finite chains of reductions are possible. □

Bibliography

M. Abadi, L. Cardelli, and P. Curien. Formal parametric polymorphism. In *Proc. of POPL'93*, pages 157–170. ACM, 1993.

B. Atkey, N. Ghani, and P. Johann. A relationally parametric model of dependent type theory. 2014.

R. Atkey, S. Lindley, and J. Yallop. Unembedding domain-specific languages. In *Proc. of Haskell '09*, pages 37–48. ACM, 2009.

H. P. Barendregt. Lambda calculi with types. *Handbook of logic in computer science*, 2:117–309, 1992.

J.-P. Bernardy and M. Lasson. Realizability and parametricity in pure type systems. In M. Hofmann, editor, *FoSSaCS*, volume 6604 of *LNCS*, pages 108–122. Springer, 2011.

J.-P. Bernardy and G. Moulin. A computational interpretation of parametricity. In *Proc. of the Symposium on Logic in Comp. Sci.* IEEE, 2012.

J.-P. Bernardy and G. Moulin. Type-theory in color. In *Proceeding of the 18th ACM SIGPLAN international conference on Funct. Programming*, 2013. To appear.

J.-P. Bernardy, P. Jansson, and R. Paterson. Parametricity and dependent types. In *Proc. of the 15th ACM SIGPLAN international conference on Funct. programming*, pages 345–356, Baltimore, Maryland, 2010. ACM. doi: 10.1145/1863543.1863592.

A. Chlipala. Parametric higher-order abstract syntax for mechanized semantics. In *Proc. of ICFP 2008*, pages 143–156. ACM, 2008.

T. Coquand and G. Huet. The calculus of constructions. Technical report, INRIA, 1986.

A. Gill, J. Launchbury, and S. Peyton Jones. A short cut to deforestation. In *Proc. of FPCA*, pages 223–232. ACM, 1993.

P. Johann. A generalization of short-cut fusion and its correctness proof. *Higher-Order and Symbol. Comp.*, 15(4):273–300, 2002.

C. Keller and M. Lasson. Parametricity in an impredicative sort. In *CSL*, pages 381–395, 2012.

N. R. Krishnaswami and D. Dreyer. Internalizing relational parametricity in the extensional calculus of constructions. In *CSL*, pages 432–451, 2013.

H. Mairson. Outline of a proof theory of parametricity. In *Proc. of FPCA 1991*, volume 523 of *LNCS*, pages 313–327. Springer-Verlag, 1991.

P. Martin-Löf. *Intuitionistic type theory*. Bibliopolis, 1984.

U. Norell. *Towards a practical programming language based on dependent type theory*. PhD thesis, Chalmers Tekniska Högskola, 2007.

C. Paulin-Mohring. Extracting Fω's programs from proofs in the calculus of constructions. In *POPL'89*, pages 89–104. ACM, 1989.

F. Pfenning and C. Paulin-Mohring. Inductively defined types in the calculus of constructions. In *MFPS*, volume 442 of *LNCS*, pages 209–228. Springer, 1990.

G. Plotkin and M. Abadi. A logic for parametric polymorphism. In *Proc. of TLCA*, volume 664 of *LNCS*, page 361–375. Springer, 1993.

N. Pouillard. Nameless, painless. In *Proc. of ICFP 2011*, ICFP '11, pages 320–332. ACM, 2011. to appear.

J. C. Reynolds. Types, abstraction and parametric polymorphism. *Information processing*, 83(1):513–523, 1983.

J. Svenningsson. *Scalable Program Analysis*. Phd thesis, Chalmers Tekniska Högskola, 2007.

The Coq development team. The Coq proof assistant, 2013.

P. Wadler. Theorems for free! In *Proc. of FPCA 1989*, pages 347–359. ACM, 1989.

P. Wadler. The Girard–Reynolds isomorphism (second edition). *Theor. Comp. Sci.*, 375(1–3):201–226, 2007.

Appendix: Additional proofs

Lemma 1. *For each term A and each variable z not free in A, we have:*

i) $\quad [\![A]\!]_{\xi,z\mapsto(z_0,z_1)} = [\![A]\!]_{\xi}$, *and*

ii) $\quad \{a\}_{\xi,z\mapsto(z_0,z_1)} \in [\![A]\!]_{\xi,z\mapsto(z_0,z_1)} = \{a\}_{\xi} \in [\![A]\!]_{\xi}$ *for all terms a.*

Proof. By simultaneous induction on the structure of the raw term A. Following the definition of our relational interpretation, we prove only i) for the case of variable, lambda, relation introduction and application; we prove ii) in the other cases, namely product, sort, and relation elimination.

Variable $[\![x_i]\!]^n \dagger^\pi$

$$\begin{aligned}
[\![[\![x_i]\!]^n \dagger^\pi]\!]_{\xi,z} &= [\![[\![x_i]\!]^n]\!]_{\xi,z} \ddagger^{\pi+1} \\
&= [\![x_i]\!]^{1+n} \ddagger^{\pi+1} \\
&= [\![[\![x_i]\!]^n]\!]_{\xi} \ddagger^{\pi+1} \\
&= [\![[\![x_i]\!]^n \dagger^\pi]\!]_{\xi}
\end{aligned}$$

Lambda $\lambda\bar{x} : \bar{A}.\, B$

$$\begin{aligned}
[\![\lambda\bar{x} : \bar{A}.\, B]\!]_{\xi,z} &= \lambda\bar{x} : [\![\bar{A}]\!]_{\xi,z}.\, [\![B]\!]_{\xi,z,x} \\
&= \lambda\bar{x} : [\![\bar{A}]\!]_{\xi}.\, [\![B]\!]_{\xi,x} && \text{by IH} \\
&= [\![\lambda\bar{x} : \bar{A}.\, B]\!]_{\xi}
\end{aligned}$$

Lambda$^\bullet$ $\lambda^\bullet\check{x} : \check{A}.\, B$

$$\begin{aligned}
[\![\lambda^\bullet\check{x} : \check{A}.\, B]\!]_{\xi,z} &= \lambda^\bullet\check{x} : [\![\check{A}]\!]_{\xi,z}.\, [\![B]\!]_{\xi,z,x} \\
&= \lambda^\bullet\check{x} : [\![\check{A}]\!]_{\xi}.\, [\![B]\!]_{\xi,x} && \text{by IH} \\
&= [\![\lambda^\bullet\check{x} : \check{A}.\, B]\!]_{\xi}
\end{aligned}$$

Application $F\,\bar{a}$

$$\begin{aligned}\llbracket F\,\bar{a}\rrbracket_{\xi,z} &= \llbracket F\rrbracket_{\xi,z}\,\llbracket \bar{a}\rrbracket_{\xi,z}\\ &= \llbracket F\rrbracket_{\xi}\,\llbracket \bar{a}\rrbracket_{\xi} && \text{by IH}\\ &= \llbracket F\,\bar{a}\rrbracket_{\xi}\end{aligned}$$

Sort s^n

$$\begin{aligned}\{a\}_{\xi,z} \in \llbracket s^n\rrbracket_{\xi,z} &= \begin{pmatrix}\{a\}_{\xi,z}\\ \cdot\end{pmatrix} \overset{\bullet}{\to} s^{1+n}\\ &= \{a\}_{\xi} \in \llbracket s^n\rrbracket_{\xi}\end{aligned}$$

Product $\forall \bar{x} : \bar{A}.\, B$

$$\begin{aligned}\{a\}_{\xi,z} \in \llbracket \forall \bar{x} : \bar{A}.\, B\rrbracket_{\xi,z} &= \forall \bar{x} : \llbracket \bar{A}\rrbracket_{\xi,z}.\, (\{a\}_{\xi,z}\,(\bar{x}/01...1)) \in \llbracket B\rrbracket_{\xi,z,x}\\ &= \forall \bar{x} : \llbracket \bar{A}\rrbracket_{\xi}.\, (\{a\}_{\xi}\,(\bar{x}/01...1)) \in \llbracket B\rrbracket_{\xi,x} && \text{by IH}\\ &= \{a\}_{\xi} \in \llbracket \forall \bar{x} : \bar{A}.\, B\rrbracket_{\xi}\end{aligned}$$

Arrow$^{\bullet}$ $\check{A} \overset{\bullet}{\to} s^n$

$$\begin{aligned}\{a\}_{\xi,z} \in \llbracket \check{A} \overset{\bullet}{\to} s^n\rrbracket_{\xi,z} &= (\llbracket \check{A}\rrbracket_{\xi,z} \oplus \{a\}_{\xi,z}) \overset{\bullet}{\to} s^{1+n}\\ &= (\llbracket \check{A}\rrbracket_{\xi} \oplus \{a\}_{\xi}) \overset{\bullet}{\to} s^{1+n} && \text{by IH}\\ &= \{a\}_{\xi} \in \llbracket \check{A} \overset{\bullet}{\to} s^n\rrbracket_{\xi}\end{aligned}$$

Application$^{\bullet}$ $F{\bullet}\check{b}$

$$\begin{aligned}\{a\}_{\xi,z} \in \llbracket F{\bullet}\check{b}\rrbracket_{\xi,z} &= \llbracket F\rrbracket_{\xi,z}{\bullet}(\llbracket \check{b}\rrbracket_{\xi} \oplus \{a\}_{\xi,z})\\ &= \llbracket F\rrbracket_{\xi}{\bullet}(\llbracket \check{b}\rrbracket_{\xi} \oplus \{a\}_{\xi}) && \text{by IH}\\ &= \{a\}_{\xi} \in \llbracket F{\bullet}\check{b}\rrbracket_{\xi}\end{aligned}$$

□

Lemma 2. *For each term $A : s^m$ and each ρ of dimension at most m, we have:*

$$\llbracket A \ddagger_{\zeta}^{\rho}\rrbracket_{\xi} = \llbracket A\rrbracket_{\xi} \ddagger_{\zeta}^{1+\rho}$$

Proof. By induction on the structure of raw term A.

Variable (1) $\llbracket x_i\rrbracket^{\,n} \dagger^{\pi}$, $x \in \xi \cap \zeta$

$$\begin{aligned}\llbracket \llbracket x_i\rrbracket^{\,n} \dagger^{\pi} \ddagger_{\zeta}^{\rho}\rrbracket_{\xi} &= \llbracket \llbracket x_i\rrbracket^{\,n} \dagger^{\pi}\rrbracket_{\xi}\\ &= \llbracket \llbracket x_i\rrbracket^{\,n}\rrbracket_{\xi} \ddagger^{\pi+1}\\ &= \llbracket x_{1i}\rrbracket^{\,n} \dagger^{\mathrm{normal}_n((0...n)\circ\pi+1)}\\ &= \llbracket x_{1i}\rrbracket^{\,n} \dagger^{\mathrm{normal}_n((0...n)\circ\pi+1)} \ddagger_{\zeta}^{\rho+1}\\ &= \llbracket \llbracket x_i\rrbracket^{\,n} \dagger^{\pi}\rrbracket_{\xi} \ddagger_{\zeta}^{\rho+1}\end{aligned}$$

Variable (2) $\llbracket x_i \rrbracket^n \dagger^\pi$, $x \notin \xi$, $x \in \zeta$

$$\begin{aligned}
\llbracket \llbracket x_i \rrbracket^n \dagger^\pi \ddagger^\rho_\zeta \rrbracket_\xi &= \llbracket \llbracket x_i \rrbracket^n \dagger^\pi \rrbracket_\xi \\
&= \llbracket \llbracket x_i \rrbracket^n \rrbracket_\xi \ddagger^{\pi+1} \\
&= \llbracket x_{0i} \rrbracket^n \dagger^{\mathrm{normal}_n((0\ldots n)\circ\pi+1)} \\
&= \llbracket x_{0i} \rrbracket^n \dagger^{\mathrm{normal}_n((0\ldots n)\circ\pi+1)} \ddagger^{\rho+1}_\zeta \\
&= \llbracket \llbracket x_i \rrbracket^n \dagger^\pi \rrbracket_\xi \ddagger^{\rho+1}_\zeta
\end{aligned}$$

Variable (3) $\llbracket x_i \rrbracket^n \dagger^\pi$, $x \in \xi$, $x \notin \zeta$

$$\begin{aligned}
\llbracket \llbracket x_i \rrbracket^n \dagger^\pi \ddagger^\rho_\zeta \rrbracket_\xi &= \llbracket \llbracket x_i \rrbracket^n \dagger^{\mathrm{normal}_n(\rho\circ\pi)} \rrbracket_\xi \\
&= \llbracket x_{1i} \rrbracket^n \dagger^{(0\ldots n)} \dagger^{1+\mathrm{normal}_n(\rho\circ\pi)} \\
&= \llbracket x_{1i} \rrbracket^n \dagger^{(0\ldots n)} \dagger^{1+\pi} \ddagger^{1+\rho}_\zeta \\
&= \llbracket \llbracket x_i \rrbracket^n \dagger^\pi \rrbracket_\xi \ddagger^{1+\rho}_\zeta
\end{aligned}$$

Variable (4) $\llbracket x_i \rrbracket^n \dagger^\pi$, $x \notin \xi$, $x \notin \zeta$

$$\begin{aligned}
\llbracket \llbracket x_i \rrbracket^n \dagger^\pi \ddagger^\rho_\zeta \rrbracket_\xi &= \llbracket \llbracket x_i \rrbracket^n \dagger^{\mathrm{normal}_n(\rho\circ\pi)} \rrbracket_\xi \\
&= \llbracket x_{0i} \rrbracket^n \dagger^{(0\ldots n)} \dagger^{1+\mathrm{normal}_n(\rho\circ\pi)} \\
&= \llbracket x_{0i} \rrbracket^n \dagger^{(0\ldots n)} \dagger^{1+\pi} \ddagger^{1+\rho}_\zeta \\
&= \llbracket \llbracket x_i \rrbracket^n \dagger^\pi \rrbracket_\xi \ddagger^{1+\rho}_\zeta
\end{aligned}$$

Lambda $\lambda \bar{x} : \bar{A}.\, B$

$$\begin{aligned}
\llbracket (\lambda \bar{x} : \bar{A}.\, B) \ddagger^\rho_\zeta \rrbracket_\xi &= \llbracket \lambda \bar{x} : \bar{A} \ddagger^\rho_\zeta . B[\bar{x} \dagger^\rho / \bar{x}] \ddagger^\rho_{\zeta,x} \rrbracket_\xi \\
&= \lambda \bar{x} : \llbracket \bar{A} \ddagger^\rho_\zeta \rrbracket_\xi . \llbracket B[\bar{x} \dagger^\rho / \bar{x}] \ddagger^\rho_{\zeta,x} \rrbracket_{\xi,x} \\
&= \lambda \bar{x} : \llbracket \bar{A} \rrbracket_\xi \ddagger^{1+\rho}_\zeta . \llbracket B[\bar{x} \dagger^\rho / \bar{x}] \rrbracket_{\xi,x} \ddagger^{1+\rho}_{\zeta,x} \qquad \text{by IH} \\
&= \lambda \bar{x} : \llbracket \bar{A} \rrbracket_\xi \ddagger^{1+\rho}_\zeta . \llbracket B \rrbracket_{\xi,x} [\bar{x} \dagger^{1+\rho} / \bar{x}] \ddagger^{1+\rho}_{\zeta,x} \\
&= (\lambda \bar{x} : \llbracket \bar{A} \rrbracket_\xi . \llbracket B \rrbracket_{\xi,x}) \ddagger^{1+\rho}_\zeta \\
&= \llbracket \lambda \bar{x} : \bar{A}.\, B \rrbracket_\xi \ddagger^{1+\rho}_\zeta
\end{aligned}$$

Lambda• $\lambda^\bullet \check{x} : \check{A}.\, B$

$$
\begin{aligned}
[\![(\lambda^\bullet \check{x} : \check{A}.\, B)\, \ddagger_\zeta^\rho]\!]_\xi &= [\![\lambda^\bullet \check{x} : \check{A}\, \ddagger_\zeta^\rho .\, B[\check{x}\dagger^\rho / \check{x}]\, \ddagger_{\zeta,x}^\rho]\!]_\xi \\
&= \lambda^\bullet \check{x} : ([\![\check{A}\, \ddagger_\zeta^\rho]\!]_\xi \oplus \{\lambda^\bullet \check{x} : \check{A}\, \ddagger_\zeta^\rho .\, B[\check{x}\dagger^\rho / \check{x}]\, \ddagger_{\zeta,x}^\rho\}_\xi). \\
&\qquad\qquad x_{01\ldots1} \in [\![B[\check{x}\dagger^\rho / \check{x}]\, \ddagger_{\zeta,x}^\rho]\!]_{\xi,x} \\
&= \lambda^\bullet \check{x} : ([\![\check{A}]\!]_\xi \oplus \{\lambda^\bullet \check{x} : \check{A}.\, B\}_\xi)\, \ddagger_\zeta^{1+\rho} . \qquad \text{by IH} \\
&\qquad (x_{01\ldots1} \in [\![B]\!]_{\xi,x})[\check{x}\dagger^{1+\rho} / \check{x}]\, \ddagger_{\zeta,x}^{1+\rho} \\
&= (\lambda^\bullet \check{x} : ([\![\check{A}]\!]_\xi \oplus \{\lambda^\bullet \check{x} : \check{A}.\, B\}_\xi).\, x_{01\ldots1} \in [\![B]\!]_{\xi,x})\, \ddagger_\zeta^{1+\rho} \\
&= [\![\lambda^\bullet \check{x} : \check{A}.\, B]\!]_\xi\, \ddagger_\zeta^{1+\rho}
\end{aligned}
$$

Application $F\,\bar{a}$

$$
\begin{aligned}
[\![(F\,\bar{a})\, \ddagger_\zeta^\rho]\!]_\xi &= [\![F\, \ddagger_\zeta^\rho\ \bar{a}\, \ddagger_\zeta^\rho]\!]_\xi \\
&= [\![F\, \ddagger_\zeta^\rho]\!]_\xi\, [\![\bar{a}\, \ddagger_\zeta^\rho]\!]_\xi \\
&= [\![F]\!]_\xi\, \ddagger_\zeta^{1+\rho}\ [\![\bar{a}]\!]_\xi\, \ddagger_\zeta^{1+\rho} \qquad \text{by IH} \\
&= [\![F\,\bar{a}]\!]_\xi\, \ddagger_\zeta^{1+\rho}
\end{aligned}
$$

Sort s^n

$$
\begin{aligned}
\{a\}_\xi\, \ddagger_\zeta^{1+\rho} \in [\![s^n\, \ddagger_\zeta^\rho]\!]_\xi &= \{a\}_\xi\, \ddagger_\zeta^{1+\rho} \in [\![s^n]\!]_\xi \\
&= \begin{pmatrix} \{a\}_\xi\, \ddagger_\zeta^{1+\rho} \\ \cdot \end{pmatrix} \xrightarrow{\bullet} s^{1+n} \\
&= (\{a\}_\xi \in [\![s^n]\!]_\xi)\, \ddagger_\zeta^{1+\rho}
\end{aligned}
$$

Product $\forall \bar{x} : \bar{A}.\, B$

$$
\begin{aligned}
&\{a\}_\xi\, \ddagger_\zeta^{1+\rho} \in [\![(\forall \bar{x} : \bar{A}.\, B)\, \ddagger_\zeta^\rho]\!]_\xi \\
&= \{a\}_\xi\, \ddagger_\zeta^{1+\rho} \in [\![\forall \bar{x} : \bar{A}\, \ddagger^\rho .\, B[\bar{x}\dagger^\rho / \bar{x}]\, \ddagger_{\zeta,x}^\rho]\!]_\xi \\
&= \forall \bar{x} : [\![\bar{A}\, \ddagger^\rho]\!]_\xi .\, (\{a\}_\xi\, \ddagger_\zeta^{1+\rho}\ (\bar{x}/01\ldots1)) \in [\![B[\bar{x}\dagger^\rho / \bar{x}]\, \ddagger_{\zeta,x}^\rho]\!]_{\xi,x} \\
&= \forall \bar{x} : [\![\bar{A}]\!]_\xi\, \ddagger^{1+\rho} .\, (\{a\}_\xi\, (\bar{x}/01\ldots1)) \in [\![B]\!]_{\xi,x})[\bar{x}\dagger^{1+\rho} / \bar{x}]\, \ddagger_\zeta^{1+\rho} \quad \text{by IH} \\
&= [\![\forall \bar{x} : \bar{A}.\, B]\!]_\xi\, \ddagger_\zeta^{1+\rho}
\end{aligned}
$$

Arrow$^\bullet$ $\check{A} \overset{\bullet}{\to} s^n$

$$
\begin{aligned}
&\{a\}_\xi \ddagger_\zeta^{1+\rho} \in [\![(\check{A} \overset{\bullet}{\to} s^n) \ddagger_\zeta^\rho]\!]_\xi \\
&= \{a\}_\xi \ddagger_\zeta^{1+\rho} \in [\![\check{A} \ddagger_\zeta^\rho \overset{\bullet}{\to} s^n]\!]_\xi \\
&= ([\![\check{A} \ddagger_\zeta^\rho]\!]_\xi \oplus \{a\}_\xi \ddagger_\zeta^{1+\rho}) \overset{\bullet}{\to} s^{1+n} \\
&= ([\![\check{A}]\!]_\xi \oplus \{a\}_\xi) \ddagger^{1+\rho} \overset{\bullet}{\to} s^{1+n} && \text{by IH} \\
&= (\{a\}_\xi \in [\![\check{A} \overset{\bullet}{\to} s^n]\!]_\xi) \ddagger_\zeta^{1+\rho}
\end{aligned}
$$

Application$^\bullet$ $F \bullet \check{b}$

$$
\begin{aligned}
\{a\}_\xi \ddagger_\zeta^{1+\rho} \in [\![(F \bullet \check{b}) \ddagger_\zeta^\rho]\!]_\xi &= \{a\}_\xi \ddagger_\zeta^{1+\rho} \in [\![F \ddagger_\zeta^\rho \bullet \check{b} \ddagger^\rho]\!]_\xi \\
&= [\![F \ddagger_\zeta^\rho]\!]_\xi \bullet ([\![\check{b} \ddagger_\zeta^\rho]\!]_\xi \oplus \{a\}_\xi \ddagger_\zeta^{1+\rho}) \\
&= [\![F]\!]_\xi \ddagger_\zeta^{1+\rho} \bullet ([\![\check{b}]\!]_\xi \oplus \{a\}_\xi) \ddagger_\zeta^{1+\rho} && \text{by IH} \\
&= (\{a\}_\xi \in [\![F \bullet \check{b}]\!]_\xi) \ddagger_\zeta^{1+\rho}
\end{aligned}
$$

□

Lemma 3. *For each term A, we have:*

$$[\![[\![A]\!]_\xi]\!]_\zeta = [\![[\![A]\!]_\zeta]\!]_\xi [\bar{x} \dagger^{(01)} / \bar{x} \mid x \in \xi \cap \zeta] \ddagger_{\xi \cap \zeta}^{(01)}$$

Proof. By structural induction on A.

Variable (1) $[\![z_i]\!]^n \dagger^\pi$, $z \in \xi$, $z \in \zeta$

$$
\begin{aligned}
[\![[\![[\![z_i]\!]^n \dagger^\pi]\!]_\xi]\!]_\zeta &= [\![[\![[\![z_i]\!]_\xi]\!]^n \ddagger^{(0\ldots n)} \ddagger^{1+\pi}]\!]_\zeta \\
&= [\![[\![[\![z_{1i}]\!]^n \dagger^{(0\ldots n)} \ddagger^{1+\pi}]\!]_\zeta \\
&= [\![[\![[\![z_{1i}]\!]^n]\!]_\zeta \ddagger^{(1\ldots 1+n)} \ddagger^{2+\pi} && \text{by Lem. 2} \\
&= [\![[\![[\![z_{1i}]\!]_\zeta]\!]^n \ddagger^{(0\ldots n)} \ddagger^{(1\ldots 1+n)} \ddagger^{2+\pi} \\
&= [\![[\![z_{11i}]\!]^n \dagger^{(0\ldots n)} \ddagger^{(1\ldots 1+n)} \ddagger^{2+\pi} \\
&= [\![[\![z_{11i}]\!]^n \dagger^{\cdots} \ddagger^{2+\pi} \\
&= [\![[\![z_{11i}]\!]^n \dagger^{\cdots} \ddagger^{(01)}_{\xi\cap\zeta} \ddagger^{2+\pi} \\
&= [\![[\![z_{11i}]\!]^n \dagger^{\cdots} \ddagger^{2+\pi} \ddagger^{(01)}_{\xi\cap\zeta} \\
&\text{since } 2+\pi \text{ and } (01) \text{ are disjoints} \\
&= [\![[\![[\![z_{1i}]\!]_\xi]\!]^n \ddagger^{(0\ldots n)} \ddagger^{(1\ldots 1+n)} \ddagger^{2+\pi} \ddagger^{(01)}_{\xi\cap\zeta} \\
&= [\![[\![[\![z_{1i}]\!]^n]\!]_\xi \ddagger^{(1\ldots 1+n)} \ddagger^{2+\pi} \ddagger^{(01)}_{\xi\cap\zeta} \\
&= [\![[\![[\![z_{1i}]\!]^n \dagger^{(0\ldots n)} \ddagger^{1+\pi}]\!]_\xi \ddagger^{(01)}_{\xi\cap\zeta} && \text{by Lem. 2} \\
&= [\![[\![[\![[\![z_i]\!]_\zeta]\!]^n \ddagger^{(0\ldots n)} \ddagger^{1+\pi}]\!]_\xi \ddagger^{(01)}_{\xi\cap\zeta} \\
&= [\![[\![[\![z_i]\!]^n \dagger^\pi]\!]_\zeta]\!]_\xi \ddagger^{(01)}_{\xi\cap\zeta} \\
&= [\![[\![[\![z_i]\!]^n \dagger^\pi]\!]_\zeta]\!]_\xi[\bar{x}\dagger^{(01)}/\bar{x} \mid x \in \xi\cap\zeta] \ddagger^{(01)}_{\xi\cap\zeta}
\end{aligned}
$$

Variable (2) $[\![z_i]\!]^n \dagger^\pi$, $z \notin \xi$, $z \in \zeta$

$$
\begin{aligned}
[\![[\![[\![z_i]\!]^n \dagger^\pi]\!]_\xi]\!]_\zeta &= [\![[\![z_i]\!]^{1+n} \dagger^{1+\pi}]\!]_\zeta \\
&= [\![[\![[\![z_i]\!]_\zeta]\!]^{1+n} \ddagger^{(0\ldots 1+n)} \ddagger^{2+\pi} \\
&= [\![[\![z_{1i}]\!]^{1+n} \dagger^{(0\ldots 1+n)} \ddagger^{2+\pi} \\
&= [\![[\![z_{1i}]\!]^{1+n} \dagger^{(1\ldots 1+n)} \ddagger^{(01)}_z \ddagger^{2+\pi} \\
&\text{since } 2+\pi \text{ and } (01) \text{ are disjoints} \\
&= [\![[\![z_{1i}]\!]^{1+n} \dagger^{(1\ldots 1+n)} \ddagger^{2+\pi} \ddagger^{(01)}_{\xi\cap\zeta} \\
&= [\![[\![z_{1i}]\!]^n \dagger^{(0\ldots n)} \ddagger^{1+\pi}]\!]_\xi \ddagger^{(01)}_{\xi\cap\zeta} \\
&= [\![[\![[\![z_i]\!]_\zeta]\!]^n \ddagger^{(0\ldots n)} \ddagger^{1+\pi}]\!]_\xi \ddagger^{(01)}_{\xi\cap\zeta} \\
&= [\![[\![[\![z_i]\!]^n \dagger^\pi]\!]_\zeta]\!]_\xi \ddagger^{(01)}_{\xi\cap\zeta} \\
&= [\![[\![[\![z_i]\!]^n \dagger^\pi]\!]_\zeta]\!]_\xi[\bar{x}\dagger^{(01)}/\bar{x} \mid x \in \xi\cap\zeta] \ddagger^{(01)}_{\xi\cap\zeta}
\end{aligned}
$$

Variable (3) $[\![z_i]\!]^n \dagger^\pi$, $z \in \xi$, $z \notin \zeta$

$$\begin{aligned}
[\![[\![[\![z_i]\!]^n \dagger^\pi]\!]_\xi]\!]_\zeta &= [\![[\![z_{1i}]\!]^n \dagger^{(0\ldots n)} \ddagger^{1+\pi}]\!]_\zeta \\
&= [\![z_{1i}]\!]^{1+n} \dagger^{(1\ldots 1+n)} \ddagger^{2+\pi} \\
&= [\![z_{1i}]\!]^{1+n} \dagger^{(0\ldots 1+n)} \ddagger_z^{(01)} \ddagger^{2+\pi} \\
&\quad \text{since } 2+\pi \text{ and } (01) \text{ are disjoints} \\
&= [\![z_{1i}]\!]^{1+n} \dagger^{(0\ldots 1+n)} \ddagger^{2+\pi} \ddagger_{\xi\cap\zeta}^{(01)} \\
&= [\![[\![z_i]\!]^{1+n} \dagger^{1+\pi}]\!]_\xi \ddagger_{\xi\cap\zeta}^{(01)} \\
&= [\![[\![[\![z_i]\!]^n \dagger^\pi]\!]_\zeta]\!]_\xi \ddagger_{\xi\cap\zeta}^{(01)} \\
&= [\![[\![[\![z_i]\!]^n \dagger^\pi]\!]_\zeta]\!]_\xi [\bar{x}\dagger^{(01)} / \bar{x} \mid x \in \xi\cap\zeta] \ddagger_{\xi\cap\zeta}^{(01)}
\end{aligned}$$

Variable (4) $[\![z_i]\!]^n \dagger^\pi$, $z \notin \xi$, $z \notin \zeta$

$$\begin{aligned}
[\![[\![[\![z_i]\!]^n \dagger^\pi]\!]_\xi]\!]_\zeta &= [\![z_i]\!]^{2+n} \dagger^{2+\pi} \\
&\quad \text{since } \dim(01) < 2 \\
&= [\![z_i]\!]^{2+n} \ddagger_{\xi\cap\zeta}^{(01)} \ddagger^{2+\pi} \\
&\quad \text{since } 2+\pi \text{ and } (01) \text{ are disjoints} \\
&= [\![z_i]\!]^{2+n} \dagger^{2+\pi} \ddagger_{\xi\cap\zeta}^{(01)} \\
&= [\![[\![[\![z_i]\!]^n \dagger^\pi]\!]_\zeta]\!]_\xi \ddagger_{\xi\cap\zeta}^{(01)} \\
&= [\![[\![[\![z_i]\!]^n \dagger^\pi]\!]_\zeta]\!]_\xi [\bar{x}\dagger^{(01)} / \bar{x} \mid x \in \xi\cap\zeta] \ddagger_{\xi\cap\zeta}^{(01)}
\end{aligned}$$

Lambda $\lambda\bar{x} : \bar{A}.\, B$

$$\begin{aligned}
[\![[\![\lambda\bar{x} : \bar{A}.\, B]\!]_\xi]\!]_\zeta &= \lambda\bar{x} : [\![[\![\bar{A}]\!]_\xi]\!]_\zeta.\, [\![[\![B]\!]_{\xi,x}]\!]_{\zeta,x} \\
&= \lambda\bar{x} : [\![[\![\bar{A}]\!]_\zeta]\!]_\xi [\bar{z}\dagger^{(01)} / \bar{z} \mid z \in \xi\cap\zeta] \ddagger_{\xi\cap\zeta}^{(01)}. \qquad \text{by IH} \\
&\qquad [\![[\![B]\!]_{\zeta,x}]\!]_{\xi,x} [\bar{z}\dagger^{(01)} / \bar{z} \mid z \in (\xi\cap\zeta)\cup\{x\}] \ddagger_{(\xi\cap\zeta)\cup\{x\}}^{(01)} \\
&= \lambda\bar{x} : [\![[\![\bar{A}]\!]_\zeta]\!]_\xi [\bar{z}\dagger^{(01)} / \bar{z} \mid z \in \xi\cap\zeta] \ddagger_{\xi\cap\zeta}^{(01)}. \\
&\qquad [\![[\![B]\!]_{\zeta,x}]\!]_{\xi,x} [\bar{z}\dagger^{(01)} / \bar{z} \mid z \in \xi\cap\zeta] \ddagger_{\xi\cap\zeta}^{(01)} \\
&= (\lambda\bar{x} : [\![[\![\bar{A}]\!]_\zeta]\!]_\xi.\, [\![[\![B]\!]_{\zeta,x}]\!]_{\xi,x}) [\bar{z}\dagger^{(01)} / \bar{z} \mid z \in \xi\cap\zeta] \ddagger_{\xi\cap\zeta}^{(01)} \\
&= [\![[\![\lambda\bar{x} : \bar{A}.\, B]\!]_\zeta]\!]_\xi [\bar{z}\dagger^{(01)} / \bar{z} \mid z \in \xi\cap\zeta] \ddagger_{\xi\cap\zeta}^{(01)}
\end{aligned}$$

Lambda• $\lambda^\bullet \check{x} : \check{A}.\, B$

$$
\begin{aligned}
&[\![[\![\lambda^\bullet \check{x} : \check{A}.\, B]\!]_{\bar{\xi}}]\!]_{\zeta} \\
&= \lambda^\bullet \check{x} : \check{C}.\, [\![[\![B]\!]_{\zeta,x}]\!]_{\bar{\xi},x}{}^{\bullet} \begin{pmatrix} x_{001\ldots1} & x_{011\ldots1} \\ x_{101\ldots1} & \cdot \end{pmatrix} \\
&= \lambda^\bullet \check{x} : \check{C}.\, [\![[\![B]\!]_{\zeta,x}]\!]_{\bar{\xi},x} [\bar{z}\,\dagger^{(01)} / \bar{z} \mid z \in (\xi \cap \zeta) \cup \{x\}]\, \ddagger^{(01)}_{(\xi\cap\zeta)\cup\{x\}}{}^{\bullet} \qquad \text{by IH} \\
&\qquad\qquad \begin{pmatrix} x_{001\ldots1} & x_{011\ldots1} \\ x_{101\ldots1} & \cdot \end{pmatrix} \\
&= \lambda^\bullet \check{x} : \check{C}'[\bar{z}\,\dagger^{(01)} / \bar{z} \mid z \in \xi \cap \zeta]\, \ddagger^{(01)}_{\xi\cap\zeta} \cdot \\
&\qquad \left([\![[\![B]\!]_{\zeta,x}]\!]_{\bar{\xi},x}{}^{\bullet} \begin{pmatrix} x_{001\ldots1} & x_{011\ldots1} \\ x_{101\ldots1} & \cdot \end{pmatrix} \right) [\bar{z}\,\dagger^{(01)} / \bar{z} \mid z \in \xi \cap \zeta]\, \ddagger^{(01)}_{\xi\cap\zeta} \\
&= \left(\lambda^\bullet \check{x} : \check{C}'.\, [\![[\![B]\!]_{\zeta,x}]\!]_{\bar{\xi},x}{}^{\bullet} \begin{pmatrix} x_{001\ldots1} & x_{011\ldots1} \\ x_{101\ldots1} & \cdot \end{pmatrix} \right) \\
&\qquad\qquad [\bar{z}\,\dagger^{(01)} / \bar{z} \mid z \in \xi \cap \zeta]\, \ddagger^{(01)}_{\xi\cap\zeta} \\
&= [\![[\![\lambda^\bullet \check{x} : \check{A}.\, B]\!]_{\zeta}]\!]_{\bar{\xi}} [\bar{z}\,\dagger^{(01)} / \bar{z} \mid z \in \xi \cap \zeta]\, \ddagger^{(01)}_{\xi\cap\zeta}
\end{aligned}
$$

Where

$$
\check{C} = \left[\begin{array}{l|l}
00i \mapsto \{\{A_i\}_{\bar{\xi}}\}_{\zeta} & 10i \mapsto [\![\{A_i\}_{\bar{\xi}}]\!]_{\zeta} \\
01i \mapsto \{[\![A_i]\!]_{\bar{\xi}}\}_{\zeta} & 11i \mapsto [\![[\![A_i]\!]_{\bar{\xi}}]\!]_{\zeta} \\
001\ldots1 \mapsto \{\{\lambda^\bullet \check{x} : \check{A}.\, B\}_{\bar{\xi}}\}_{\zeta} & 101\ldots1 \mapsto [\![\{\lambda^\bullet \check{x} : \check{A}.\, B\}_{\bar{\xi}}]\!]_{\zeta} \\
011\ldots1 \mapsto \{\lambda^\bullet \check{x} : ([\![\check{A}]\!]_{\bar{\xi}} \oplus \{\lambda^\bullet \check{x} : \check{A}.\, B\}_{\bar{\xi}}).\, x_{01\ldots1} \in [\![B]\!]_{\bar{\xi},x}\}_{\zeta} &
\end{array} \right]
$$

$$
= \check{C}'[\bar{z}\,\dagger^{(01)} / \bar{z} \mid z \in \xi \cap \zeta]\, \ddagger^{(01)}_{\xi\cap\zeta} \quad \text{by IH}
$$

and

$$
\check{C}' = \left[\begin{array}{l|l}
00i \mapsto \{\{A_i\}_{\zeta}\}_{\bar{\xi}} & 10i \mapsto [\![\{A_i\}_{\zeta}]\!]_{\bar{\xi}} \\
01i \mapsto \{[\![A_i]\!]_{\zeta}\}_{\bar{\xi}} & 11i \mapsto [\![[\![A_i]\!]_{\zeta}]\!]_{\bar{\xi}} \\
001\ldots1 \mapsto \{\{\lambda^\bullet \check{x} : \check{A}.\, B\}_{\zeta}\}_{\bar{\xi}} & 101\ldots1 \mapsto [\![\{\lambda^\bullet \check{x} : \check{A}.\, B\}_{\zeta}]\!]_{\bar{\xi}} \\
011\ldots1 \mapsto \{\lambda^\bullet \check{x} : ([\![\check{A}]\!]_{\zeta} \oplus \{\lambda^\bullet \check{x} : \check{A}.\, B\}_{\zeta}).\, x_{01\ldots1} \in [\![B]\!]_{\zeta,x}\}_{\bar{\xi}} &
\end{array} \right]
$$

Application $F\,\bar{a}$

$$
\begin{aligned}
[\![[\![F\,\bar{a}]\!]_{\bar{\xi}}]\!]_{\zeta} &= [\![[\![F]\!]_{\bar{\xi}}]\!]_{\zeta}\, [\![[\![\bar{a}]\!]_{\bar{\xi}}]\!]_{\zeta} \\
&= [\![[\![F]\!]_{\zeta}]\!]_{\bar{\xi}} [\bar{z}\,\dagger^{(01)} / \bar{z} \mid z \in \xi \cap \zeta]\, \ddagger^{(01)}_{\xi\cap\zeta} \qquad \text{by IH} \\
&\qquad [\![[\![\bar{a}]\!]_{\zeta}]\!]_{\bar{\xi}} [\bar{z}\,\dagger^{(01)} / \bar{z} \mid z \in \xi \cap \zeta]\, \ddagger^{(01)}_{\xi\cap\zeta} \\
&= [\![[\![F\,\bar{a}]\!]_{\zeta}]\!]_{\bar{\xi}} [\bar{z}\,\dagger^{(01)} / \bar{z} \mid z \in \xi \cap \zeta]\, \ddagger^{(01)}_{\xi\cap\zeta}
\end{aligned}
$$

Sort s^n

$$\begin{aligned}\{\{a\}_{\vec{\xi}}\}_{\zeta} \in [\![[\![s^n]\!]_{\vec{\xi}}]\!]_{\zeta} &= \begin{pmatrix} \{\{a\}_{\vec{\xi}}\}_{\zeta} & \{[\![a]\!]_{\vec{\xi}}\}_{\zeta} \\ [\![\{a\}_{\vec{\xi}}]\!]_{\zeta} & \cdot \end{pmatrix} \overset{\bullet}{\to} s^{n+2} \\ &= \left(\begin{pmatrix} \{\{a\}_{\zeta}\}_{\xi} & \{[\![a]\!]_{\zeta}\}_{\xi} \\ [\![\{a\}_{\zeta}]\!]_{\xi} & \cdot \end{pmatrix} \overset{\bullet}{\to} s^{n+2} \right) \\ &\qquad [\bar{z}\dagger^{(01)}/\bar{z} \mid z \in \xi \cap \zeta]\, \ddagger^{(01)}_{\xi\cap\zeta} \\ &= (\{\{a\}_{\zeta}\}_{\xi} \in [\![[\![s^n]\!]_{\zeta}]\!]_{\xi}[\bar{z}\dagger^{(01)}/\bar{z} \mid z \in \xi \cap \zeta])\, \ddagger^{(01)}_{\xi\cap\zeta}\end{aligned}$$

Product $\forall \bar{x} : \bar{A}.\, B$

$$\begin{aligned}&\{\{a\}_{\vec{\xi}}\}_{\zeta} \in [\![[\![\forall \bar{x} : \bar{A}.\, B]\!]_{\vec{\xi}}]\!]_{\zeta} \\ &= \forall \bar{x} : [\![[\![\bar{A}]\!]_{\vec{\xi}}]\!]_{\zeta}.\, [\![[\![\bar{B}]\!]_{\vec{\xi},x}]\!]_{\zeta,x} \begin{pmatrix} \{\{a\}_{\vec{\xi}}\}_{\zeta}(\bar{x}/001...1) & \{[\![a]\!]_{\vec{\xi}}\}_{\zeta}(\bar{x}/011...1) \\ [\![\{a\}_{\vec{\xi}}]\!]_{\zeta}(\bar{x}/101...1) & \cdot \end{pmatrix} \\ &= \forall \bar{x} : [\![[\![\bar{A}]\!]_{\vec{\xi}}]\!]_{\zeta}[\bar{z}\dagger^{(01)}/\bar{z} \mid z \in \xi \cap \zeta]\, \ddagger^{(01)}_{\xi\cap\zeta}. \qquad \text{by IH} \\ &\quad \left([\![[\![\bar{B}]\!]_{\vec{\xi},x}]\!]_{\zeta,x} \begin{pmatrix} \{\{a\}_{\vec{\xi}}\}_{\zeta}(\bar{x}/001...1) & \{[\![a]\!]_{\vec{\xi}}\}_{\zeta}(\bar{x}/011...1) \\ [\![\{a\}_{\vec{\xi}}]\!]_{\zeta}(\bar{x}/101...1) & \cdot \end{pmatrix} \right) \\ &\qquad [\bar{z}\dagger^{(01)}/\bar{z} \mid z \in \xi \cap \zeta \cup \{x\}]\, \ddagger^{(01)}_{\xi\cap\zeta\cup\{x\}} \\ &= (\{\{a\}_{\zeta}\}_{\xi} \in [\![[\![\forall \bar{x} : \bar{A}.\, B]\!]_{\zeta}]\!]_{\xi}[\bar{z}\dagger^{(01)}/\bar{z} \mid z \in \xi \cap \zeta])\, \ddagger^{(01)}_{\xi\cap\zeta}\end{aligned}$$

Arrow$^{\bullet}$ $\check{A} \overset{\bullet}{\to} s^n$

$$\begin{aligned}&\{\{a\}_{\vec{\xi}}\}_{\zeta} \in [\![[\![\check{A} \overset{\bullet}{\to} s^n]\!]_{\vec{\xi}}]\!]_{\zeta} \\ &= \check{C} \overset{\bullet}{\to} s^{2+n} \\ &= (\check{C}' \overset{\bullet}{\to} s^{2+n})[\bar{z}\dagger^{(01)}/\bar{z} \mid z \in \xi \cap \zeta])\, \ddagger^{(01)}_{\xi\cap\zeta} \\ &= (\{\{a\}_{\zeta}\}_{\xi} \in [\![[\![\check{A} \overset{\bullet}{\to} s^n]\!]_{\zeta}]\!]_{\xi}[\bar{z}\dagger^{(01)}/\bar{z} \mid z \in \xi \cap \zeta])\, \ddagger^{(01)}_{\xi\cap\zeta}\end{aligned}$$

Where

$$\check{C} = \left[\begin{array}{l|l} 00i \mapsto \{\{A_i\}_{\vec{\xi}}\}_{\zeta} & 10i \mapsto [\![\{A_i\}_{\vec{\xi}}]\!]_{\zeta} \\ 01i \mapsto \{[\![A_i]\!]_{\vec{\xi}}\}_{\zeta} & 11i \mapsto [\![[\![A_i]\!]_{\vec{\xi}}]\!]_{\zeta} \\ 001...1 \mapsto \{\{a\}_{\vec{\xi}}\}_{\zeta} & 101...1 \mapsto [\![\{a\}_{\vec{\xi}}]\!]_{\zeta} \\ 011...1 \mapsto \{[\![a]\!]_{\vec{\xi}}\}_{\zeta} & \end{array}\right]$$

$$= \check{C}'[\bar{z}\dagger^{(01)}/\bar{z} \mid z \in \xi \cap \zeta]\, \ddagger^{(01)}_{\xi\cap\zeta} \quad \text{by IH}$$

and

$$\check{C}' = \left[\begin{array}{l|l} 00i \mapsto \{\{A_i\}_{\zeta}\}_{\vec{\xi}} & 10i \mapsto [\![\{A_i\}_{\zeta}]\!]_{\vec{\xi}} \\ 01i \mapsto \{[\![A_i]\!]_{\zeta}\}_{\vec{\xi}} & 11i \mapsto [\![[\![A_i]\!]_{\zeta}]\!]_{\vec{\xi}} \\ 001...1 \mapsto \{\{a\}_{\zeta}\}_{\vec{\xi}} & 101...1 \mapsto [\![\{a\}_{\zeta}]\!]_{\vec{\xi}} \\ 011...1 \mapsto \{[\![a]\!]_{\zeta}\}_{\vec{\xi}} & \end{array}\right]$$

Application$^\bullet$ $F{\bullet}\check{A}$

$$\{\{a\}_{\vec{\xi}}\}_{\vec{\zeta}} \in [\![[\![F{\bullet}\check{A}]\!]_{\vec{\xi}}]\!]_{\vec{\zeta}} = [\![[\![F]\!]_{\vec{\xi}}]\!]_{\vec{\zeta}}{\bullet}\check{C}$$
$$= (\{\{a\}_{\vec{\zeta}}\}_{\vec{\xi}} \in [\![F{\bullet}\check{A}]\!]_{\vec{\xi}}[\bar{z}\dagger^{(01)}/\bar{z} \mid z \in \vec{\xi} \cap \vec{\zeta}])\ddagger^{(01)}_{\vec{\xi}\cap\vec{\zeta}}$$

Where

$$\check{C} = \left[\begin{array}{l|l} 00i \mapsto \{\{A_i\}_{\vec{\xi}}\}_{\vec{\zeta}} & 10i \mapsto [\![\{A_i\}_{\vec{\xi}}]\!]_{\vec{\zeta}} \\ 01i \mapsto \{[\![A_i]\!]_{\vec{\xi}}\}_{\vec{\zeta}} & 11i \mapsto [\![[\![A_i]\!]_{\vec{\xi}}]\!]_{\vec{\zeta}} \\ 001...1 \mapsto \{\{a\}_{\vec{\xi}}\}_{\vec{\zeta}} & 101...1 \mapsto [\![\{a\}_{\vec{\xi}}]\!]_{\vec{\zeta}} \\ 011...1 \mapsto \{[\![a]\!]_{\vec{\xi}}\}_{\vec{\zeta}} & \end{array}\right]$$
$$= \check{C}'[\bar{z}\dagger^{(01)}/\bar{z} \mid z \in \vec{\xi} \cap \vec{\zeta}]\ddagger^{(01)}_{\vec{\xi}\cap\vec{\zeta}} \quad \text{by IH}$$

and

$$\check{C}' = \left[\begin{array}{l|l} 00i \mapsto \{\{A_i\}_{\vec{\zeta}}\}_{\vec{\xi}} & 10i \mapsto [\![\{A_i\}_{\vec{\zeta}}]\!]_{\vec{\xi}} \\ 01i \mapsto \{[\![A_i]\!]_{\vec{\zeta}}\}_{\vec{\xi}} & 11i \mapsto [\![[\![A_i]\!]_{\vec{\zeta}}]\!]_{\vec{\xi}} \\ 001...1 \mapsto \{\{a\}_{\vec{\zeta}}\}_{\vec{\xi}} & 101...1 \mapsto [\![\{a\}_{\vec{\zeta}}]\!]_{\vec{\xi}} \\ 011...1 \mapsto \{[\![a]\!]_{\vec{\zeta}}\}_{\vec{\xi}} & \end{array}\right]$$

□

Lemma 4 ($[\![\cdot]\!]$ and substitution, part 1)**.** *For each term A, and each variable z not in $\vec{\xi}$, we have:*

i) $[\![A[u/z_i]]\!]_{\vec{\xi}} = [\![A]\!]_{\vec{\xi},z}[\{u\}_{\vec{\xi}}/z_{0i}][[\![u]\!]_{\vec{\xi}}/z_{1i}]$, *and*

ii) $\{a[u/z_i]\}_{\vec{\xi}} \in [\![A[u/z_i]]\!]_{\vec{\xi}} = (\{a\}_{\vec{\xi},z} \in [\![A]\!]_{\vec{\xi},z})[\{u\}_{\vec{\xi}}/z_{0i}][[\![u]\!]_{\vec{\xi}}/z_{1i}]$.

Proof. By simultaneous induction on the structure of A.

Variable (1) $[\![z_i]\!]^n\dagger^\pi$

$$[\![[\![z_i]\!]^n\dagger^\pi[u/z_i]]\!]_{\vec{\xi}} = [\![[\![u]\!]^n\ddagger^\pi]\!]_{\vec{\xi}}$$
$$= [\![[\![u]\!]^n]\!]_{\vec{\xi}}\ddagger^{\pi+1} \quad \text{by Lem. 2}$$
$$= [\![[\![u]\!]_{\vec{\xi}}]\!]^n\ddagger^{(0\ldots n)}\ddagger^{\pi+1} \quad \text{by Cor. 1}$$
$$= [\![z_{1i}]\!]^n\ddagger^{(0\ldots n)}\ddagger^{\pi+1}[\{u\}_{\vec{\xi}}/z_{0i}][[\![u]\!]_{\vec{\xi}}/z_{1i}]$$
$$= [\![[\![z_i]\!]_{\vec{\xi},z}]\!]^n\ddagger^{(0\ldots n)}\ddagger^{\pi+1}[\{u\}_{\vec{\xi}}/z_{0i}][[\![u]\!]_{\vec{\xi}}/z_{1i}]$$
$$= [\![[\![z_i]\!]^n]\!]_{\vec{\xi},z}\ddagger^{\pi+1}[\{u\}_{\vec{\xi}}/z_{0i}][[\![u]\!]_{\vec{\xi}}/z_{1i}]$$
$$= [\![[\![z_i]\!]^n\dagger^\pi]\!]_{\vec{\xi},z\mapsto(z_0,z_1)}[\{u\}_{\vec{\xi}}/z_{0i}][[\![u]\!]_{\vec{\xi}}/z_{1i}] \quad \text{by Lem. 2}$$

Variable (2) $[\![x_j]\!]^n\dagger^\pi$, with $x \neq z$

$$[\![[\![x_j]\!]^n\dagger^\pi[u/z_i]]\!]_{\vec{\xi}} = [\![[\![x_j]\!]^n\dagger^\pi]\!]_{\vec{\xi}}$$
$$= [\![[\![x_j]\!]^n\dagger^\pi]\!]_{\vec{\xi}}[\{u\}_{\vec{\xi}}/z_{0i}][[\![u]\!]_{\vec{\xi}}/z_{1i}]$$
$$= [\![[\![x_j]\!]^n\dagger^\pi]\!]_{\vec{\xi},z\mapsto(z_0,z_1)}[\{u\}_{\vec{\xi}}/z_{0i}][[\![u]\!]_{\vec{\xi}}/z_{1i}] \quad \text{by Lem. 1}$$

Lambda $\lambda \bar{x} : \bar{A}.\, B$

$$\begin{aligned}
[\![(\lambda \bar{x} : \bar{A}.\, B)[u/z_i]]\!]_{\bar{\xi}} &= [\![\lambda \bar{x} : \bar{A}[u/z_i].\, B[u/z_i]]\!]_{\bar{\xi}} \\
&= \lambda \bar{x} : [\![\bar{A}[u/z_i]]\!]_{\bar{\xi}}.\, [\![B[u/z_i]]\!]_{\bar{\xi},x} \\
&= \lambda \bar{x} : [\![\bar{A}]\!]_{\bar{\xi},z}[\{u\}_{\bar{\xi}}/z_{0i}][[\![u]\!]_{\bar{\xi}}/z_{1i}]. \qquad \text{by IH} \\
&\qquad\qquad [\![B]\!]_{\bar{\xi},x,z}[\{u\}_{\bar{\xi}}/z_{0i}][[\![u]\!]_{\bar{\xi}}/z_{1i}] \\
&= [\![\lambda \bar{x} : \bar{A}.\, B]\!]_{\bar{\xi},z\mapsto(z_0,z_1)}[\{u\}_{\bar{\xi}}/z_{0i}][[\![u]\!]_{\bar{\xi}}/z_{1i}]
\end{aligned}$$

Lambda• $\lambda^\bullet \check{x} : \check{A}.\, B$

$$\begin{aligned}
&[\![(\lambda^\bullet \check{x} : \check{A}.\, B)[u/z_i]]\!]_{\bar{\xi}} \\
&= [\![\lambda^\bullet \check{x} : \check{A}[u/z_i].\, B[u/z_i]]\!]_{\bar{\xi}} \\
&= \lambda^\bullet \check{x} : ([\![\check{A}[u/z_i]]\!]_{\bar{\xi}} \oplus \{\lambda^\bullet \check{x} : \check{A}[u/z_i].\, B[u/z_i]\}_{\bar{\xi}}).\, x_{01\ldots1} \in [\![B[u/z_i]]\!]_{\bar{\xi},x} \\
&= \lambda^\bullet \check{x} : ([\![\check{A}]\!]_{\bar{\xi}} \oplus \{\lambda^\bullet \check{x} : \check{A}[u/z_i].\, B[u/z_i]\}_{\bar{\xi}}). \\
&\qquad\qquad x_{01\ldots1} \in [\![B[u/z_i]]\!]_{\bar{\xi},x} \\
&= \lambda^\bullet \check{x} : ([\![\check{A}]\!]_{\bar{\xi},z} \oplus \lambda^\bullet \check{x} : \check{A}.\, B)[\{u\}_{\bar{\xi}}/z_{0i}][[\![u]\!]_{\bar{\xi}}/z_{1i}]. \qquad \text{by IH} \\
&\qquad\qquad (x_{01\ldots1} \in [\![B]\!]_{\bar{\xi},x,z})[\{u\}_{\bar{\xi}}/z_{0i}][[\![u]\!]_{\bar{\xi}}/z_{1i}] \\
&= (\lambda^\bullet \check{x} : ([\![\check{A}]\!]_{\bar{\xi},z} \oplus \{\lambda^\bullet \check{x} : \check{A}.\, B\}_{\bar{\xi},z}).\, x_{01\ldots1} \in [\![B]\!]_{\bar{\xi},x,z}) \\
&\qquad\qquad [\{u\}_{\bar{\xi}}/z_{0i}][[\![u]\!]_{\bar{\xi}}/z_{1i}] \\
&= [\![\lambda^\bullet \check{x} : \check{A}.\, B]\!]_{\bar{\xi},z}[\{u\}_{\bar{\xi}}/z_{0i}][[\![u]\!]_{\bar{\xi}}/z_{1i}]
\end{aligned}$$

Application $F\,\bar{a}$

$$\begin{aligned}
[\![(F\,\bar{a})[u/z_i]]\!]_{\bar{\xi}} &= [\![F[u/z_i]\,\bar{a}[u/z_i]]\!]_{\bar{\xi}} \\
&= [\![F[u/z_i]]\!]_{\bar{\xi}}\, [\![\bar{a}[u/z_i]]\!]_{\bar{\xi}} \\
&= [\![F]\!]_{\bar{\xi},z}[\{u\}_{\bar{\xi}}/z_{0i}][[\![u]\!]_{\bar{\xi}}/z_{1i}]\, [\![\bar{a}]\!]_{\bar{\xi},z}[\{u\}_{\bar{\xi}}/z_{0i}][[\![u]\!]_{\bar{\xi}}/z_{1i}] \quad \text{by IH} \\
&= ([\![F]\!]_{\bar{\xi},z}\, [\![\bar{a}]\!]_{\bar{\xi},z})[\{u\}_{\bar{\xi}}/z_{0i}][[\![u]\!]_{\bar{\xi}}/z_{1i}] \\
&= [\![F\,\bar{a}]\!]_{\bar{\xi},z}[\{u\}_{\bar{\xi}}/z_{0i}][[\![u]\!]_{\bar{\xi}}/z_{1i}]
\end{aligned}$$

Sort s^n

$$\begin{aligned}
\{a[u/z_i]\}_{\bar{\xi}} \in [\![s^n[u/z_i]]\!]_{\bar{\xi}} &= \begin{pmatrix} \{a\}_{\bar{\xi},z}[\{u\}_{\bar{\xi}}/z_{0i}] \\ \cdot \end{pmatrix} \xrightarrow{\bullet} s^{1+n} \\
&= (\{a\}_{\bar{\xi},z} \in [\![s^n]\!]_{\bar{\xi},z})[\{u\}_{\bar{\xi}}/z_{0i}][[\![u]\!]_{\bar{\xi}}/z_{1i}]
\end{aligned}$$

Product $\forall \bar{x} : \bar{A}.\, B$

$$\begin{aligned}
&\{a[u/z_i]\}_{\bar{\xi}} \in [\![(\forall \bar{x} : \bar{A}.\, B)[u/z_i]]\!]_{\bar{\xi}} \\
&= \forall \bar{x} : [\![\bar{A}[u/z_i]]\!]_{\bar{\xi}}.\, (\{a[u/z_i]\}_{\bar{\xi}}\, (\bar{x}/01\ldots1)) \in [\![B[u/z_i]]\!]_{\bar{\xi},x,z} \\
&= (\forall \bar{x} : [\![\bar{A}]\!]_{\bar{\xi},z}.\, (\{a\}_{\bar{\xi},z}\, (\bar{x}/01\ldots1)) \in [\![B]\!]_{\bar{\xi},x,z})[\{u\}_{\bar{\xi}}/z_{0i}][[\![u]\!]_{\bar{\xi}}/z_{1i}] \quad \text{by IH} \\
&= (\{a\}_{\bar{\xi},z} \in [\![\forall \bar{x} : \bar{A}.\, B]\!]_{\bar{\xi},z})[\{u\}_{\bar{\xi}}/z_{0i}][[\![u]\!]_{\bar{\xi}}/z_{1i}]
\end{aligned}$$

Arrow• $\check{A} \overset{\bullet}{\to} s^n$

$$\begin{aligned}
&\{a[u/z_i]\}_\xi \in [\![(\check{A} \overset{\bullet}{\to} s^n)[u/z_i]]\!]_\xi \\
&= ([\![\check{A}[u/z_i]]\!]_\xi \oplus \{a[u/z_i]\}_\xi) \overset{\bullet}{\to} s^{1+n} \\
&= (([\![\check{A}]\!]_{\xi,z} \oplus \{a\}_{\xi,z}) \overset{\bullet}{\to} s^{1+n})[\{u\}_\xi/z_{0i}][[\![u]\!]_\xi/z_{1i}] \qquad \text{by IH} \\
&= (\{a\}_{\xi,z} \in [\![\check{A} \overset{\bullet}{\to} s^n]\!]_{\xi,z})[\{u\}_\xi/z_{0i}][[\![u]\!]_\xi/z_{1i}]
\end{aligned}$$

Application• $F \bullet \check{b}$

$$\begin{aligned}
&\{a[u/z_i]\}_\xi \in [\![(F \bullet \check{b})[u/z_i]]\!]_\xi \\
&= [\![F[u/z_i]]\!]_\xi \bullet ([\![\check{b}[u/z_i]]\!]_\xi \oplus \{a[u/z_i]\}_\xi) \\
&= [\![F]\!]_{\xi,z}[\{u\}_\xi/z_{0i}][[\![u]\!]_\xi/z_{1i}] \bullet ([\![\check{b}]\!]_{\xi,z} \oplus \{a\}_{\xi,z})[\{u\}_\xi/z_{0i}][[\![u]\!]_\xi/z_{1i}] \quad \text{by IH} \\
&= (\{a\}_{\xi,z} \in [\![F \bullet \check{b}]\!]_{\xi,z})[\{u\}_\xi/z_{0i}][[\![u]\!]_\xi/z_{1i}]
\end{aligned}$$

□

Lemma 5 ($[\![\cdot]\!]$ and substitution, part 2)**.** *For each term A, for variable z not free in A or contained in ξ, we have:*

i) $[\![A[u/z_i]]\!]_\xi = [\![A]\!]_\xi[\{u\}_\xi/z_{0i}]$, *and*

ii) $\{a[u/z_i]\}_\xi \in [\![A[u/z_i]]\!]_\xi = (\{a\}_\xi \in [\![A]\!]_\xi)[\{u\}_\xi/z_{0i}]$.

Proof. By simultaneous induction on the structure of A.

Variable (1) $[\![z_i]\!]^n \dagger^\pi$ Impossible.

Variable (2) $[\![x_j]\!]^n \dagger^\pi$, with $x \neq z$

$$\begin{aligned}
[\![[\![x_j]\!]^n \dagger^\pi[u/z_i]]\!]_\xi &= [\![[\![x_j]\!]^n \dagger^\pi]\!]_\xi \\
&= [\![[\![x_j]\!]^n \dagger^\pi]\!]_\xi[\{u\}_\xi/z_{0i}]
\end{aligned}$$

Lambda $\lambda\bar{x} : \bar{A}.\, B$

$$\begin{aligned}
[\![(\lambda\bar{x} : \bar{A}.\, B)[u/z_i]]\!]_\xi &= [\![\lambda\bar{x} : \bar{A}[u/z_i].\, B[u/z_i]]\!]_\xi \\
&= \lambda\bar{x} : [\![\bar{A}[u/z_i]]\!]_\xi.\, [\![B[u/z_i]]\!]_{\xi,x} \\
&= \lambda\bar{x} : [\![\bar{A}]\!]_\xi[\{u\}_\xi/z_{0i}].\, [\![B]\!]_{\xi,x}[\{u\}_\xi/z_{0i}] \quad \text{by IH} \\
&= [\![\lambda\bar{x} : \bar{A}.\, B]\!]_\xi[\{u\}_\xi/z_{0i}]
\end{aligned}$$

Lambda• $\lambda^\bullet \check{x} : \check{A}.\, B$

$$
\begin{aligned}
&[\![(\lambda^\bullet \check{x} : \check{A}.\, B)[u/z_i]]\!]_{\check{\xi}} \\
&= [\![\lambda^\bullet \check{x} : \check{A}[u/z_i].\, B[u/z_i]]\!]_{\check{\xi}} \\
&= \lambda^\bullet \check{x} : ([\![\check{A}[u/z_i]]\!]_{\check{\xi}} \oplus \{\lambda^\bullet \check{x} : \check{A}[u/z_i].\, B[u/z_i]\}_{\check{\xi}}).\, x_{01\ldots1} \in [\![B[u/z_i]]\!]_{\check{\xi},x} \\
&= \lambda^\bullet \check{x} : ([\![\check{A}]\!]_{\check{\xi}} \oplus \{\lambda^\bullet \check{x} : \check{A}.\, B\}_{\check{\xi}})[\{u\}_{\check{\xi}}/z_{0i}].\ (x_{01\ldots1} \in [\![B]\!]_{\check{\xi},x})[\{u\}_{\check{\xi}}/z_{0i}] && \text{by IH} \\
&= [\![\lambda^\bullet \check{x} : \check{A}.\, B]\!]_{\check{\xi}}[\{u\}_{\check{\xi}}/z_{0i}]
\end{aligned}
$$

Application $F\,\bar{a}$

$$
\begin{aligned}
[\![(F\,\bar{a})[u/z_i]]\!]_{\check{\xi}} &= [\![F[u/z_i]\ \bar{a}[u/z_i]]\!]_{\check{\xi}} \\
&= [\![F[u/z_i]]\!]_{\check{\xi}}\ [\![\bar{a}[u/z_i]]\!]_{\check{\xi}} \\
&= [\![F]\!]_{\check{\xi}}[\{u\}_{\check{\xi}}/z_{0i}]\ [\![\bar{a}]\!]_{\check{\xi}}[\{u\}_{\check{\xi}}/z_{0i}] && \text{by IH} \\
&= ([\![F]\!]_{\check{\xi}}\ [\![\bar{a}]\!]_{\check{\xi}})[\{u\}_{\check{\xi}}/z_{0i}] \\
&= [\![F\,\bar{a}]\!]_{\check{\xi}}[\{u\}_{\check{\xi}}/z_{0i}]
\end{aligned}
$$

Sort s^n

$$
\begin{aligned}
\{a[u/z_i]\}_{\check{\xi}} \in [\![s^n[u/z_i]]\!]_{\check{\xi}} &= \begin{pmatrix} \{a\}_{\check{\xi}}[\{u\}_{\check{\xi}}/z_{0i}] \\ \cdot \end{pmatrix} \overset{\bullet}{\to} s^{1+n} \\
&= (\{a\}_{\check{\xi}} \in [\![s^n]\!]_{\check{\xi}})[\{u\}_{\check{\xi}}/z_{0i}]
\end{aligned}
$$

Product $\forall \bar{x} : \bar{A}.\, B$

$$
\begin{aligned}
&\{a[u/z_i]\}_{\check{\xi}} \in [\![(\forall \bar{x} : \bar{A}.\, B)[u/z_i]]\!]_{\check{\xi}} \\
&= \forall \bar{x} : [\![\bar{A}[u/z_i]]\!]_{\check{\xi}}.\, (\{a[u/z_i]\}_{\check{\xi}}\ (\bar{x}/01\ldots1)) \in [\![B[u/z_i]]\!]_{\check{\xi},x} \\
&= (\forall \bar{x} : [\![\bar{A}]\!]_{\check{\xi}}.\, (\{a\}_{\check{\xi}}\ (\bar{x}/01\ldots1)) \in [\![B]\!]_{\check{\xi},x})[\{u\}_{\check{\xi}}/z_{0i}] && \text{by IH} \\
&= (\{a\}_{\check{\xi}} \in [\![\forall \bar{x} : \bar{A}.\, B]\!]_{\check{\xi}})[\{u\}_{\check{\xi}}/z_{0i}]
\end{aligned}
$$

Arrow• $\check{A} \overset{\bullet}{\to} s^n$

$$
\begin{aligned}
&\{a[u/z_i]\}_{\check{\xi}} \in [\![(\check{A} \overset{\bullet}{\to} s^n)[u/z_i]]\!]_{\check{\xi}} \\
&= ([\![\check{A}[u/z_i]]\!]_{\check{\xi}} \oplus \{a[u/z_i]\}_{\check{\xi}}) \overset{\bullet}{\to} s^{1+n} \\
&= (([\![\check{A}]\!]_{\check{\xi}} \oplus \{a\}_{\check{\xi}}) \overset{\bullet}{\to} s^{1+n})[\{u\}_{\check{\xi}}/z_{0i}] && \text{by IH} \\
&= (\{a\}_{\check{\xi}} \in [\![\check{A} \overset{\bullet}{\to} s^n]\!]_{\check{\xi}})[\{u\}_{\check{\xi}}/z_{0i}]
\end{aligned}
$$

Application• $F{\bullet}\check{b}$

$$
\begin{aligned}
&\{a[u/z_i]\}_{\check{\xi}} \in [\![(F{\bullet}\check{b})[u/z_i]]\!]_{\check{\xi}} \\
&= [\![F[u/z_i]]\!]_{\check{\xi}}{\bullet}([\![\check{b}[u/z_i]]\!]_{\check{\xi}} \oplus \{a[u/z_i]\}_{\check{\xi}}) \\
&= [\![F]\!]_{\check{\xi}}[\{u\}_{\check{\xi}}/z_{0i}][[\![u]\!]_{\check{\xi}}/z_{1i}]{\bullet}([\![\check{b}]\!]_{\check{\xi}} \oplus \{a\}_{\check{\xi}})[\{u\}_{\check{\xi}}/z_{0i}] && \text{by IH} \\
&= (\{a\}_{\check{\xi}} \in [\![F{\bullet}\check{b}]\!]_{\check{\xi}})[\{u\}_{\check{\xi}}/z_{0i}]
\end{aligned}
$$

□

Lemma 6 (Symmetry)**.** *For each term A, $[\![A]\!]^n$ is symmetric in its n first dimensions. More specifically,*

i) $\quad [\![A]\!]^n_\xi \ddagger^\pi_\xi = [\![A]\!]^n_\xi \ddagger^{\mathrm{normal}_n(\pi)}_\xi$, *and*

ii) $\quad (a \in [\![A]\!]^n_\xi) \ddagger^\pi_\xi = (a \in [\![A]\!]^n_\xi) \ddagger^{\mathrm{normal}_n(\pi)}_\xi$

Proof. By simultaneous induction on the structure of A.

Variable (1) $[\![x_i]\!]^m \dagger^\rho$, $x \in \xi$

$$\begin{aligned}
[\![[\![x_i]\!]^m \dagger^\rho]\!]^n_\xi \ddagger^\pi_\xi &= [\![x_{1\dots1i}]\!]^m \dagger^{n+(0\dots m)} \ddagger^{n+\rho} \ddagger^\pi_\xi \\
&= [\![x_{1\dots1i}]\!]^m \dagger^{n+(0\dots m)} \ddagger^{n+\rho} \\
&= [\![[\![x_i]\!]^m \dagger^\rho]\!]^n_\xi \ddagger^{\mathrm{normal}_n(\pi)}_\xi
\end{aligned}$$

Variable (2) $[\![x_i]\!]^m \dagger^\rho$, $x \notin \xi$

$$\begin{aligned}
[\![[\![x_i]\!]^m \dagger^\rho]\!]^n_\xi \ddagger^\pi_\xi &= [\![x_i]\!]^{n+m} \dagger^{n+\rho} \ddagger^\pi_\xi \\
&= [\![x_i]\!]^{n+m} \dagger^{\mathrm{normal}_{n+m}((n+\rho)\circ\pi)} \\
&\quad \text{since } n+\rho \text{ and } (0 \dots n-1) \text{ are disjoint} \\
&= [\![x_i]\!]^{n+m} \dagger^{\mathrm{normal}_{n+m}((n+\rho)\circ\mathrm{normal}_n(\pi))} \\
&= [\![[\![x_i]\!]^m \dagger^\rho]\!]^n_\xi \ddagger^{\mathrm{normal}_n(\pi)}_\xi
\end{aligned}$$

Lambda (Lambda$^\bullet$ is similar) $\lambda\bar{x} : \bar{A}.\, B$

$$\begin{aligned}
[\![\lambda\bar{x} : \bar{A}.\, B]\!]^n_\xi \ddagger^\pi_\xi &= \lambda\bar{x} : [\![\bar{A}]\!]^n_\xi \ddagger^\pi_\xi . [\![B]\!]^n_{\xi,x} [\bar{x} \dagger^\pi / \bar{x}] \ddagger^\pi_{\xi,x} \\
&= \lambda\bar{x} : [\![\bar{A}]\!]^n_\xi \ddagger^{\mathrm{normal}_n(\pi)}_\xi . [\![B]\!]^n_{\xi,x} [\bar{x} \dagger^\pi / \bar{x}] \ddagger^{\mathrm{normal}_n(\pi)}_{\xi,x} && \text{by IH} \\
&= \lambda\bar{x} : [\![\bar{A}]\!]^n_\xi \ddagger^{\mathrm{normal}_n(\pi)}_\xi . [\![B]\!]^n_{\xi,x} [\bar{x} \dagger^{\mathrm{normal}_n(\pi)} / \bar{x}] \ddagger^{\mathrm{normal}_n(\pi)}_{\xi,x} \\
&= [\![\lambda\bar{x} : \bar{A}.\, B]\!]^n_\xi \ddagger^{\mathrm{normal}_n(\pi)}_\xi
\end{aligned}$$

Application (Application$^\bullet$ is similar) $F\,\bar{a}$

$$\begin{aligned}
[\![F\,\bar{a}]\!]^n_\xi \ddagger^\pi_\xi &= \lambda\bar{x} : [\![\bar{F}]\!]^n_\xi \ddagger^\pi_\xi . [\![a]\!]^n_\xi \ddagger^\pi_\xi \\
&= \lambda\bar{x} : [\![\bar{F}]\!]^n_\xi \ddagger^{\mathrm{normal}_n(\pi)}_\xi . [\![a]\!]^n_\xi \ddagger^{\mathrm{normal}_n(\pi)}_{\xi,x} && \text{by IH} \\
&= [\![\lambda\bar{x} : \bar{A}.\, B]\!]^n_\xi \ddagger^{\mathrm{normal}_n(\pi)}_\xi
\end{aligned}$$

Product (Arrow$^\bullet$ is similar) $\forall \bar{x} : \bar{A}.\, B$

$$
\begin{aligned}
&(C \in [\![\forall \bar{x} : \bar{A}.\, B]\!]^n_{\tilde{\xi}})\, \ddagger^{\pi}_{\tilde{\xi}} \\
&= \forall \bar{x} : [\![\bar{A}]\!]^n_{\tilde{\xi}}\, \ddagger^{\pi}_{\tilde{\xi}}\, .\, (C\, \bar{x} \in [\![B]\!]^n_{\tilde{\xi},x})[\bar{x} \dagger^{\pi} / \bar{x}]\, \ddagger^{\pi}_{\tilde{\xi},x} \\
&= \forall \bar{x} : [\![\bar{A}]\!]^n_{\tilde{\xi}}\, \ddagger^{\mathrm{normal}_n(\pi)}_{\tilde{\xi}}\, .\, (C\, \bar{x} \in [\![B]\!]^n_{\tilde{\xi},x})[\bar{x} \dagger^{\pi} / \bar{x}]\, \ddagger^{\mathrm{normal}_n(\pi)}_{\tilde{\xi},x} && \text{by IH} \\
&= \forall \bar{x} : [\![\bar{A}]\!]^n_{\tilde{\xi}}\, \ddagger^{\mathrm{normal}_n(\pi)}_{\tilde{\xi}}\, .\, (C\, \bar{x} \in [\![B]\!]^n_{\tilde{\xi},x})[\bar{x} \dagger^{\mathrm{normal}_n(\pi)} / \bar{x}]\, \ddagger^{\mathrm{normal}_n(\pi)}_{\tilde{\xi},x} \\
&= (C \in [\![\forall \bar{x} : \bar{A}.\, B]\!]^n_{\tilde{\xi}})\, \ddagger^{\mathrm{normal}_n(\pi)}_{\tilde{\xi}}
\end{aligned}
$$

Sort s

The result $(C \in [\![s]\!]^n_{\tilde{\xi}})\, \ddagger^{\pi}_{\tilde{\xi}} = (C \in [\![s]\!]^n_{\tilde{\xi}})\, \ddagger^{\mathrm{normal}_n(\pi)}_{\tilde{\xi}}$ stems from an easy induction on n. □

Lemma 12 (Generalized abstraction)**.** *Assuming that* $\tilde{\xi}$ *conforms to* Γ*,*

i) $\Gamma \vdash A : B \;\Rightarrow\; [\![\Gamma]\!]_{\tilde{\xi}} \vdash [\![A]\!]_{\tilde{\xi}} : \{A\}_{\tilde{\xi}} \in [\![B]\!]_{\tilde{\xi}}$

ii) $\Gamma \vdash A : B \;\Rightarrow\; [\![\Gamma]\!]_{\tilde{\xi}} \vdash \{A\}_{\tilde{\xi}} : \{B\}_{\tilde{\xi}}$

iii) $\Gamma \vdash B : s^n \;\Rightarrow\; [\![\Gamma]\!]_{\tilde{\xi}}, x : B \vdash x \in [\![B]\!]_{\tilde{\xi}} : s^{n+1}$

Proof. The lemmas are proved by transforming derivation trees. They mutually depend on each other, (but only for structurally smaller statements, hence the recursion is sound). For each lemma, each rule is treated. The rule being handled is written before the corresponding part of the resulting derivation.

In the proofs, the application of each sub-lemma to an arbitrary derivation $\Gamma \vdash A : B$ are written as follows:

i) $[\![\Gamma \vdash A : B]\!]_{\tilde{\xi}}$

ii) $|\Gamma \vdash A : B|_{\tilde{\xi}}$

iii) $\{\Gamma \vdash A : B\}_{\tilde{\xi}}$

We only give further details for the two first items in the following; iii) stemming from simple application of induction hypotheses.

i) $\Gamma \vdash A : B \Rightarrow [\![\Gamma]\!]_{\tilde{\xi}} \vdash [\![A]\!]_{\tilde{\xi}} : \{A\}_{\tilde{\xi}} \in [\![B]\!]_{\tilde{\xi}}$

Axiom; Rel-Elim; Rel-Form; Product In this case, the definition of $[\![A]\!]_{\tilde{\xi}}$ falls through: a new relation is introduced. The proof relies on the next sub-lemma.

$$\Gamma \vdash A : s$$

$$\dfrac{\dfrac{\vdots\,|\Gamma \vdash A : s|_{\xi} \qquad \vdots\,\{\Gamma \vdash A : s\}_{\xi}}{[\![\Gamma]\!]_{\xi}, z_0 : A \vdash z_0 \in [\![A]\!]_{\xi} : s \quad [\![\Gamma]\!]_{\xi} \vdash A : s}\,\text{Rel-I}}{[\![\Gamma]\!]_{\xi} \vdash \lambda^{\bullet} \tilde{z} : A.\, z_0 \in [\![A]\!]_{\xi} : A \xrightarrow{\bullet} s}\,\text{DEF}}{[\![\Gamma]\!]_{\xi} \vdash [\![A]\!]_{\xi} : A \in [\![s]\!]_{\xi}}$$

Weakening

$$\dfrac{\Gamma \vdash A : B \quad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B}\,\text{WK}$$

-$x \notin \xi$

$$\dfrac{\dfrac{\dfrac{\vdots\,[\![\Gamma \vdash A : B]\!]_{\xi} \qquad \vdots\,\{\Gamma \vdash C : s\}_{\xi}}{[\![\Gamma]\!]_{\xi} \vdash [\![A]\!]_{\xi} : A \in [\![B]\!]_{\xi} \qquad [\![\Gamma]\!]_{\xi} \vdash C : s}\,\text{WK}}{[\![\Gamma]\!]_{\xi}, x : C \vdash [\![A]\!]_{\xi} : A \in [\![B]\!]_{\xi}}\,\text{DEF}}{[\![\Gamma, x : C]\!]_{\xi} \vdash [\![A]\!]_{\xi} : A \in [\![B]\!]_{\xi}}$$

$-x \in \xi$

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\ [\![\Gamma \vdash C : s]\!]_\xi \\ [\![\Gamma]\!]_\xi \vdash [\![C]\!]_\xi : C \in [\![s]\!]_\xi\end{array} \quad \begin{array}{c}\vdots\ \{\Gamma \vdash C : s\}_\xi \\ [\![\Gamma]\!]_\xi \vdash C : s\end{array}}{[\![\Gamma]\!]_\xi, x_0 : C \vdash [\![C]\!]_\xi : C \in [\![s]\!]_\xi}\,\text{WK}}{[\![\Gamma]\!]_\xi, x_0 : C \vdash [\![C]\!]_\xi : C \overset{\bullet}{\to} s}\,\text{DEF} \qquad \dfrac{\begin{array}{c}\vdots\ \{\Gamma \vdash C : s\}_\xi \\ [\![\Gamma]\!]_\xi \vdash C : s\end{array}}{[\![\Gamma]\!]_\xi, x_0 : C \vdash x_0 : C}\,\text{ST}}{[\![\Gamma]\!]_\xi, x_0 : C \vdash [\![C]\!]_\xi \bullet x_0 : s}\,\text{APP}$$

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\ [\![\Gamma \vdash A : B]\!]_\xi \\ [\![\Gamma]\!]_\xi \vdash [\![A]\!]_\xi : A \in [\![B]\!]_\xi\end{array} \quad \begin{array}{c}\vdots\ \{\Gamma \vdash C : s\}_\xi \\ [\![\Gamma]\!]_\xi \vdash C : s\end{array}}{[\![\Gamma]\!]_\xi, x_0 : C \vdash [\![A]\!]_\xi : A \in [\![B]\!]_\xi}\,\text{WK} \quad \vdots}{[\![\Gamma]\!]_\xi, \bar{x} : \begin{pmatrix} C \\ [\![C]\!]_\xi \end{pmatrix} \vdash [\![A]\!]_\xi : A \in [\![B]\!]_\xi}\,\text{WK}}{[\![\Gamma, x : C]\!]_\xi \vdash [\![A]\!]_\xi : A \in [\![B]\!]_\xi}\,\text{DEF}$$

Rel-Intro

$$\frac{\Gamma, \check{z} : \check{A} \vdash B : s \quad \Gamma \vdash \check{A} : s}{\Gamma \vdash (\lambda^\bullet \check{z} : \check{A}.\, B) : \check{A} \overset{\bullet}{\to} s}\,\text{Rel-I}$$

$$\frac{\dfrac{\dfrac{\vdots\ |\Gamma, \check{z} : \check{A} \vdash B : s|_\xi}{[\![\Gamma]\!]_\xi, \check{z} : \check{A}, z_{01\ldots1} : B \vdash z_{01\ldots1} \in [\![B]\!]_\xi : s}\,\text{WK}}{[\![\Gamma]\!]_\xi, \check{z} : ([\![\check{A}]\!]_\xi \oplus (\lambda^\bullet \check{z} : \check{A}.\, B)) \vdash z_{01\ldots1} \in [\![B]\!]_\xi : s} \quad \frac{\vdots}{[\![\Gamma]\!]_\xi \vdash ([\![\check{A}]\!]_\xi \oplus (\lambda^\bullet \check{z} : \check{A}.\, B)) : s}}{\dfrac{[\![\Gamma]\!]_\xi \vdash (\lambda^\bullet \check{z} : ([\![\check{A}]\!]_\xi \oplus (\lambda^\bullet \check{z} : \check{A}.\, B)).\, z_{01\ldots1} \in [\![B]\!]_\xi) : (([\![\check{A}]\!]_\xi \oplus (\lambda^\bullet \check{z} : \check{A}.\, B)) \overset{\bullet}{\to} s)}{[\![\Gamma]\!]_\xi \vdash [\![\lambda^\bullet \check{z} : \check{A}.\, B]\!]_\xi : (\lambda^\bullet \check{z} : \check{A}.\, B \in [\![\check{A} \overset{\bullet}{\to} s]\!]_\xi)}\,\text{DEF}}\,\text{Rel-I}$$

Application

$$\frac{\Gamma \vdash F : (\forall \bar{x} : \bar{A}.\, B) \quad \Gamma \vdash \bar{a} : \bar{A}}{\Gamma \vdash F\ \bar{a} : B[x \mapsto \bar{a}]}\,\text{APP}$$

$$\frac{\dfrac{\dfrac{\dfrac{\vdots\ [\![\Gamma \vdash F : (\forall \bar{x} : \bar{A}.\, B)]\!]_\xi}{[\![\Gamma]\!]_\xi \vdash [\![F]\!]_\xi : F \in [\![\forall \bar{x} : \bar{A}.\, B]\!]_\xi}\,\text{DEF}}{[\![\Gamma]\!]_\xi \vdash [\![F]\!]_\xi : (\forall \bar{x} : [\![\bar{A}]\!]_\xi.\, (F\ \bar{x}) \in [\![B]\!]_{\xi,x})} \quad \frac{\vdots}{[\![\Gamma]\!]_\xi \vdash [\![\bar{a}]\!]_\xi : [\![\bar{A}]\!]_\xi}}{[\![\Gamma]\!]_\xi \vdash [\![F]\!]_\xi\ [\![\bar{a}]\!]_\xi : ((F\ \bar{x}) \in [\![B]\!]_{\xi,x})[\bar{x} \mapsto \bar{a}]}\,\text{APP}}{[\![\Gamma]\!]_\xi \vdash [\![F]\!]_\xi\ [\![\bar{a}]\!]_\xi : (F\ \bar{a}) \in [\![B[\bar{x} \mapsto \bar{a}]]\!]_\xi}\,\text{Lem. 4}}{[\![\Gamma]\!]_\xi \vdash [\![F\ \bar{a}]\!]_\xi : (F\ \bar{a}) \in [\![B[\bar{x} \mapsto \bar{a}]]\!]_\xi}\,\text{DEF}$$

Abstraction

$$\frac{\Gamma, z : \bar{A} \vdash b : B \quad \Gamma \vdash (\forall \bar{x} : \bar{A}.\, B) : s}{\Gamma \vdash (\lambda \bar{z} : \bar{A}.\, b) : (\forall \bar{x} : \bar{A}.\, B)}\,\text{ABS}$$

$$\dfrac{\dfrac{\dfrac{\vdots\ [\![\Gamma, z : \bar{A} \vdash b : B]\!]_{\xi,z}}{[\![\Gamma, z : \bar{A}]\!]_{\xi,z} \vdash [\![b]\!]_{\xi,z} : b \in [\![B]\!]_{\xi,z}}\,\text{DEF}}{[\![\Gamma]\!]_\xi, z : [\![\bar{A}]\!]_\xi \vdash [\![b]\!]_{\xi,z} : b \in [\![B]\!]_{\xi,z}} \qquad \{[\![\Gamma]\!]_\xi \vdash (\forall \bar{z} : [\![\bar{A}]\!]_\xi . b \in [\![B]\!]_{\xi,z}) : s\}}\,\text{ABS}}{[\![\Gamma]\!]_\xi \vdash (\lambda \bar{z} : [\![\bar{A}]\!]_\xi . [\![b]\!]_{\xi,z}) : (\forall \bar{z} : [\![\bar{A}]\!]_\xi . b \in [\![B]\!]_{\xi,z})}\,\text{DEF}}{[\![\Gamma]\!]_\xi \vdash [\![\lambda \bar{z} : \bar{A}. b]\!]_\xi : (\lambda \bar{z} : \bar{A}. b) \in [\![\forall \bar{z} : \bar{A}. B]\!]_\xi}\,\text{DEF}$$

Conversion

$$\dfrac{\Gamma \vdash A : B' \quad \Gamma \vdash B : s \quad B' =_\beta B}{\Gamma \vdash A : B}\,\text{CONV}$$

$$\dfrac{\dfrac{\vdots\ |\Gamma \vdash B : s|_\xi}{[\![\Gamma]\!]_\xi, x : B \vdash x \in [\![B]\!]_\xi : s} \qquad \dfrac{\vdots\ \{\Gamma \vdash A : B'\}_\xi \qquad \vdots\ \{\Gamma \vdash B : s\}_\xi}{[\![\Gamma]\!]_\xi \vdash A : B' \qquad [\![\Gamma]\!]_\xi \vdash B : s \quad B' =_\beta B}\,\text{CONV}}{[\![\Gamma]\!]_\xi \vdash A \in [\![B]\!]_\xi : s}\,\text{subst}$$

$$\dfrac{\dfrac{\vdots\ [\![\Gamma \vdash A : B']\!]_\xi}{[\![\Gamma]\!]_\xi \vdash [\![A]\!]_\xi : A \in [\![B']\!]_\xi} \qquad \dfrac{B' =_\beta B}{A \in [\![B']\!]_\xi =_\beta A \in [\![B]\!]_\xi}\,\text{Lem. 9}}{[\![\Gamma]\!]_\xi \vdash [\![A]\!]_\xi : A \in [\![B]\!]_\xi}\,\text{CONV}$$

Start

$$\dfrac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}\,\text{ST}$$

-$x \notin \xi$ Since ξ conforms to Γ, no variable of ξ is in Γ.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}\text{ST}}{\Gamma, x : A \vdash [\![x]\!]^d : x \in [\![A]\!]}\text{PARAM}}{[\![\Gamma]\!]_\xi, x : A \vdash [\![x]\!]^d : x \in [\![A]\!]_\xi}\text{(conforms)}}{[\![\Gamma, x : A]\!]_\xi \vdash [\![x]\!]_\xi : x \in [\![A]\!]_\xi}\text{DEF}$$

-$x \in \xi$

$$\dfrac{\dfrac{\begin{array}{c}\vdots |\Gamma \vdash A : s|_\xi \\ [\![\Gamma]\!]_\xi, x_0 : A \vdash x_0 \in [\![A]\!]_\xi : s\end{array}}{[\![\Gamma]\!]_\xi, x_0 : A, x_1 : x_0 \in [\![A]\!]_\xi \vdash x_1 : x_0 \in [\![A]\!]_\xi}\text{ST}}{[\![\Gamma, x : A]\!]_\xi \vdash [\![x]\!]_\xi : x \in [\![A]\!]_\xi}\text{DEF}$$

Param

$$\dfrac{\Gamma \vdash x : A}{\Gamma \vdash [\![x]\!]^d : x \in [\![A]\!]}\text{PARAM}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\llbracket\Gamma\rrbracket_\xi \vdash \llbracket x\rrbracket^{\,d} \in \llbracket x \in \llbracket A\rrbracket\rrbracket_\xi : s}}{\llbracket\Gamma\rrbracket_\xi \vdash \llbracket\llbracket x\rrbracket^{\,d}\rrbracket_\xi : (\llbracket x\rrbracket_\xi \in \llbracket x \in \llbracket A\rrbracket_\xi\rrbracket)}\,\text{DEF} \qquad \dfrac{\dfrac{\dfrac{\vdots}{\llbracket\Gamma \vdash x : A\rrbracket_\xi}}{\llbracket\Gamma\rrbracket_\xi \vdash \llbracket x\rrbracket_\xi : x \in \llbracket A\rrbracket_\xi}\,\text{PARAM}}{\llbracket\Gamma\rrbracket_\xi \vdash \llbracket\llbracket x\rrbracket_\xi\rrbracket^{\,d+1} : \llbracket x\rrbracket_\xi \in \llbracket x \in \llbracket A\rrbracket_\xi\rrbracket}\,\text{DEF}}{\llbracket\Gamma\rrbracket_\xi \vdash \llbracket\llbracket x\rrbracket^{\,d}\rrbracket_\xi : (\llbracket x\rrbracket_\xi \in \llbracket x \in \llbracket A\rrbracket_\xi\rrbracket) \;=_\beta\; (\llbracket x\rrbracket^{\,d} \in \llbracket x \in \llbracket A\rrbracket\rrbracket_\xi)}\,\text{CONV}$$

$$\dfrac{\dfrac{\dfrac{}{\llbracket\llbracket A\rrbracket\rrbracket_\xi =_\beta \llbracket\llbracket A\rrbracket_\xi\rrbracket}\,\text{Eq. (7)}}{\llbracket\llbracket A\rrbracket\rrbracket_\xi \bullet \begin{pmatrix} x & \llbracket x\rrbracket \\ \llbracket x\rrbracket_\xi & \cdot \end{pmatrix} =_\beta (\llbracket\llbracket A\rrbracket_\xi\rrbracket \bullet \begin{pmatrix} x & \llbracket x\rrbracket \\ \llbracket x\rrbracket_\xi & \cdot \end{pmatrix})}\,\beta\text{-Rel-Elim}}{(\llbracket x\rrbracket^{\,d} \in \llbracket x \in \llbracket A\rrbracket\rrbracket_\xi) =_\beta (\llbracket x\rrbracket_\xi \in \llbracket x \in \llbracket A\rrbracket_\xi\rrbracket)}\,\text{DEF}$$

ii) $\Gamma \vdash B : s^n \Rightarrow \llbracket\Gamma\rrbracket_\xi, x : \{B\}_\xi \vdash x \in \llbracket B\rrbracket_\xi : s^{n+1}$

Axiom

$$\dfrac{}{\vdash s_1 : s_2}\,\text{AX}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash s_1 : s_2}\text{AX}}{x : s_1 \vdash x : s_1}\text{ST}}{x : s_1 \vdash x : s_1}\text{DEF}}{x : s_1 \vdash x \xrightarrow{\bullet} s_1 : s_2}\text{Rel-F}}{x : s_1 \vdash x \in [\![s_1]\!]_\xi : s_2}\text{DEF}$$

Start

$$\dfrac{\Gamma \vdash s : s_2}{\Gamma, y : s \vdash y : s}\text{ST}$$

$y \notin \xi$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash s : s_2}{\Gamma, y : s \vdash y : s}\text{ST}}{\Gamma, y : s \vdash [\![y]\!]^d : y \in [\![s]\!]}\text{PARAM} \quad \dfrac{\Gamma \vdash s : s_2}{\Gamma, y : s \vdash y : s}\text{ST}}{\Gamma, y : s, x : y \vdash [\![y]\!]^d : y \in [\![s]\!]}\text{WK}}{\Gamma, y : s, x : y \vdash [\![y]\!]^d : y \xrightarrow{\bullet} s}\text{DEF} \quad \dfrac{\dfrac{\Gamma \vdash s : s_2}{\Gamma, y : s \vdash y : s}\text{ST}}{\Gamma, y : s, x : y \vdash x : y}\text{ST}}{\Gamma, y : s, x : y \vdash [\![y]\!]^d \bullet x : s}\text{Rel-E}}{[\![\Gamma]\!]_\xi, y : s, x : y \vdash [\![y]\!]^d \bullet x : s}\text{(conforms)}}{[\![\Gamma, y : s]\!]_\xi, x : y \vdash x \in [\![y]\!]_\xi : s}\text{DEF}$$

$\text{-}y \in \xi$

$$\dfrac{\dfrac{\dfrac{\vdots}{\Gamma, y : s, y_1 : y \overset{\bullet}{\to} s, x : y \vdash y_1 : y \overset{\bullet}{\to} s} \quad \dfrac{\vdots}{\Gamma, y : s, y_1 : y \overset{\bullet}{\to} s \vdash y : s}\,\text{ST} \quad \Gamma, y : s, y_1 : y \overset{\bullet}{\to} s, x : y \vdash x : y}{[\![\Gamma]\!]_\xi, y : s, y_1 : y \overset{\bullet}{\to} s, x : y \vdash y_1 \bullet x : s}\,\text{Rel-E}}{[\![\Gamma, y : s]\!]_\xi, x : y \vdash x \in [\![y]\!]_\xi : s}\,\text{DEF}$$

Weakening

$$\dfrac{\Gamma \vdash B : s \quad \Gamma \vdash C : s}{\Gamma, y : C \vdash B : s}\,\text{WK}$$

$\text{-}y \notin \xi$

$$\dfrac{\dfrac{\dfrac{\vdots\, |\Gamma \vdash B : s|_\xi \quad \vdots\, \{\Gamma \vdash C : s\}_\xi}{[\![\Gamma]\!]_\xi, x : B \vdash x \in [\![B]\!]_\xi : s \quad [\![\Gamma]\!]_\xi \vdash C : s}\,\text{THINNING}}{[\![\Gamma]\!]_\xi, y : C, x : B \vdash x \in [\![B]\!]_\xi : s}\,\text{DEF}}{[\![\Gamma, y : C]\!]_\xi, x : B \vdash x \in [\![B]\!]_\xi : s}$$

$\text{-}y \in \xi$

$$\dfrac{\dfrac{\dfrac{\vdots\, |\Gamma \vdash B : s|_\xi \quad \vdots\, \{\Gamma \vdash C : s\}_\xi \quad \vdots\, |\Gamma \vdash C : s|_\xi}{[\![\Gamma]\!]_\xi, x : B \vdash [\![A]\!]_\xi : A \in [\![B]\!]_\xi \quad [\![\Gamma]\!]_\xi \vdash C : s \quad [\![\Gamma]\!]_\xi, y : C \vdash y \in [\![C]\!]_\xi : s}\,\text{THINNING}}{[\![\Gamma]\!]_\xi, y : C, y_1 : y \in [\![C]\!]_\xi, x : B \vdash x \in [\![B]\!]_\xi : s}\,\text{DEF}}{[\![\Gamma, y : C]\!]_\xi, x : B \vdash x \in [\![B]\!]_\xi : s}$$

Rel-Elim

$$\dfrac{\Gamma \vdash F : \check{A} \overset{\bullet}{\to} s \quad \Gamma \vdash \check{a} : \check{A}}{\Gamma \vdash F \bullet \check{a} : s}\ \text{Rel-E}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots\ [\![\Gamma \vdash F : \check{A} \overset{\bullet}{\to} s]\!]_\xi \qquad \vdots\ \{\Gamma \vdash F \bullet \check{a} : s\}_\xi}{[\![\Gamma]\!]_\xi \vdash [\![F]\!]_\xi : F \in [\![\check{A} \overset{\bullet}{\to} s]\!]_\xi \qquad [\![\Gamma]\!]_\xi \vdash F \bullet \check{a} : s}\ \text{WK}}{[\![\Gamma]\!]_\xi, z_0 : F \bullet \check{a} \vdash [\![F]\!]_\xi : F \in [\![\check{A} \overset{\bullet}{\to} s]\!]_\xi}\ \text{DEF}}{[\![\Gamma]\!]_\xi, z_0 : F \bullet \check{a} \vdash [\![F]\!]_\xi : ([\![\check{A}]\!]_\xi \oplus F) \overset{\bullet}{\to} s} \qquad \dfrac{\dfrac{\vdots\ [\![\Gamma]\!]_\xi, z_0 : F \bullet \check{a} \vdash [\![\check{a} : \check{A}]\!]_\xi \qquad \dfrac{\dfrac{\vdots\ \{\Gamma \vdash F \bullet \check{a} : s\}_\xi}{[\![\Gamma]\!]_\xi, z_0 \vdash F \bullet \check{a} : s}\ \text{ST}}{[\![\Gamma]\!]_\xi, z_0 : F \bullet \check{a} \vdash z_0 : F \bullet \check{a}}\ \text{DEF}}{[\![\Gamma]\!]_\xi, z_0 : F \bullet \check{a} \vdash z_0 : F \bullet ([\![\check{a}]\!]_\xi /\!/ 01...1)}\ \text{DEF}}{[\![\Gamma]\!]_\xi, z_0 : F \bullet \check{a} \vdash ([\![\check{a}]\!]_\xi \oplus z_0) : ([\![\check{A}]\!]_\xi \oplus F)}\ \text{Rel-E}}{[\![\Gamma]\!]_\xi, z_0 : F \bullet \check{a} \vdash [\![F]\!]_\xi \bullet ([\![\check{a}]\!]_\xi \oplus z_0) : s}\ \text{DEF}}{[\![\Gamma]\!]_\xi, z_0 : F \bullet \check{a} \vdash z_0 \in [\![F \bullet \check{a}]\!]_\xi : s}\ \text{DEF}$$

Rel-Intro Absurd: the type is a relation ($\check{A} \overset{\bullet}{\to} s^n$), which cannot be a sort.

Rel-Form

$$\dfrac{\Gamma \vdash \check{A} : s_1}{\Gamma \vdash (\check{A} \overset{\bullet}{\to} s_1) : s_2}\ \text{Rel-F}$$

$$\dfrac{\dfrac{\dfrac{\vdots\ [\![\Gamma]\!]_\xi, z_0 : (\check{A} \overset{\bullet}{\to} s_1) \vdash [\![\check{A} : s_1]\!]_\xi \qquad \vdots\ [\![\Gamma]\!]_\xi, z_0 : (\check{A} \overset{\bullet}{\to} s_1) \vdash z_0 : \check{A} \overset{\bullet}{\to} s_1}{[\![\Gamma]\!]_\xi, z_0 : (\check{A} \overset{\bullet}{\to} s_1) \vdash ([\![\check{A}]\!]_\xi \oplus z_0) : s_1}\ \text{DEF}}{[\![\Gamma]\!]_\xi, z_0 : (\check{A} \overset{\bullet}{\to} s_1) \vdash ([\![\check{A}]\!]_\xi \oplus z_0) \overset{\bullet}{\to} s_1 : s_2}\ \text{Rel-F}}{[\![\Gamma]\!]_\xi, z_0 : (\check{A} \overset{\bullet}{\to} s_1) \vdash z_0 \in [\![\check{A} \overset{\bullet}{\to} s_1]\!]_\xi : s_2}\ \text{DEF}$$

Application

$$\dfrac{\dfrac{\Gamma \vdash \bar{A} : s_1}{\Gamma \vdash F : \bar{A} \to s}\,\text{gen} \quad \Gamma \vdash \bar{a} : \bar{A}}{\Gamma \vdash F\ \bar{A} : s}\,\text{APP}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots\ [\![\Gamma \vdash F : \bar{A} \to s : s]\!]_\xi}{[\![\Gamma]\!]_\xi \vdash [\![F]\!]_\xi : F \in [\![\bar{A} \to s]\!]_\xi}\,\text{DEF} \quad \vdots\ [\![\Gamma \vdash \bar{a} : \bar{A} : s]\!]_\xi}{[\![\Gamma]\!]_\xi \vdash [\![F]\!]_\xi : \bar{a} \in [\![\bar{A}]\!]_\xi \to F\ \bar{a} \in [\![s]\!]_\xi} \quad [\![\Gamma]\!]_\xi \vdash [\![\bar{a}]\!]_\xi : \bar{a} \in [\![\bar{A}]\!]_\xi}\,\text{APP}}{[\![\Gamma]\!]_\xi \vdash [\![F]\!]_\xi\ [\![\bar{a}]\!]_\xi : F\ \bar{a} \in [\![s]\!]_\xi}\,\text{DEF}}{[\![\Gamma]\!]_\xi \vdash [\![F]\!]_\xi\ [\![\bar{a}]\!]_\xi : F\ \bar{a} \overset{\bullet}{\to} s} \quad \dfrac{\dfrac{\vdots\ \{\Gamma \vdash F\ \bar{a} : s\}_\xi}{[\![\Gamma]\!]_\xi \vdash F\ \bar{a} : s}\,\text{ST}}{[\![\Gamma]\!]_\xi, x : F\ \bar{a} \vdash x : F\ \bar{a}}\,\text{Rel-E}}{[\![\Gamma]\!]_\xi, x : F\ \bar{a} \vdash [\![F]\!]_\xi\ [\![\bar{a}]\!]_\xi\ x : s}\,\text{DEF}}{[\![\Gamma]\!]_\xi, x : F\ \bar{a} \vdash x \in [\![F\ \bar{a}]\!]_\xi : s}\,\text{DEF}$$

Abstraction Absurd.

Product

$$\dfrac{\Gamma \vdash \bar{A} : s_1 \quad \Gamma, x : \bar{A} \vdash B : s_2}{\Gamma \vdash (\forall \bar{x} : \bar{A}.\, B) : s_3}\,(s_1, s_2, s_3)$$

$$\dfrac{\dfrac{[\![\Gamma]\!]_\xi, f : (\forall \bar{x} : \bar{A}.\, B) \vdash [\![\bar{A}]\!]_\xi : s_1 \quad \dfrac{[\![\Gamma]\!]_\xi, f : (\forall \bar{x} : \bar{A}.\, B), x : [\![\bar{A}]\!]_\xi \vdash f\ x_0 : B \quad \dfrac{\dfrac{\vdots\ |\Gamma, x : \bar{A} \vdash B : s_2|_{\xi,x}}{[\![\Gamma]\!]_\xi, x : [\![\bar{A}]\!]_\xi, z : B \vdash z \in [\![B]\!]_{\xi,x} : s_2}\,\text{WK}}{[\![\Gamma]\!]_\xi, f : (\forall \bar{x} : \bar{A}.\, B), x : [\![\bar{A}]\!]_\xi, z : B \vdash z \in [\![B]\!]_{\xi,x} : s_2}\,\text{subst}}{[\![\Gamma]\!]_\xi, f : (\forall \bar{x} : \bar{A}.\, B), x : [\![\bar{A}]\!]_\xi \vdash (f\ x_0) \in [\![B]\!]_{\xi,x} : s_2}\,(s_1, s_2, s_3)}{[\![\Gamma]\!]_\xi, f : (\forall \bar{x} : \bar{A}.\, B) \vdash (\forall \bar{x} : [\![\bar{A}]\!]_\xi.\, (f\ x_0) \in [\![B]\!]_{\xi,x}) : s_3}\,\text{DEF}}{[\![\Gamma]\!]_\xi, f : (\forall \bar{x} : \bar{A}.\, B) \vdash f \in [\![\forall \bar{x} : \bar{A}.\, B]\!]_\xi : s_3}\,\text{DEF}$$

Conversion

$$\frac{\Gamma \vdash B : s \quad \Gamma \vdash s : s \quad s =_\beta s}{\Gamma \vdash B : s}\ \text{CONV}$$

Trivial.

Param Absurd: the type of the parametricity witness is $z \in [\![B]\!]_{\vec{\zeta}}$, which cannot be a sort s^n. □

Theorem 7 (Soundness)**.** *If* $\Gamma \vdash_{\mathcal{P}} A : B$*, then*

$$\langle\!|\Gamma|\!\rangle \vdash_{\mathcal{O}} \langle\!|A|\!\rangle : \langle\!|B|\!\rangle .$$

Proof. We proceed by induction on the derivation; however the proof requires a stronger induction hypothesis when the derivation $\Gamma \vdash_{\mathcal{P}} A : B$ starts with the APPLICATION rule, hence we generalize the statement as follows:

> Let $\Gamma \vdash_{\mathcal{P}} A : B : s^n$, k such that $k \leq \epsilon(x)$ for each free variable x, and $\pi \in \mathfrak{S}_{n+k}$. Then
>
> $$\langle\!|\Gamma|\!\rangle \vdash_{\mathcal{O}} \langle\!|\, [\![A]\!]^k \ddagger^\pi \,|\!\rangle : \langle\!|\, (A \in [\![B]\!]^k) \ddagger^\pi \,|\!\rangle \qquad (9)$$

However, for the sake of readability we only prove the specialized statement

$$\Gamma \vdash A : B \Longrightarrow \langle\!|\Gamma|\!\rangle \vdash_{\mathcal{O}} \langle\!|A|\!\rangle : \langle\!|B|\!\rangle$$

(The proof for equation (9) stems from an additional decreasing induction on $k \leq \cap_{x \text{ free}} \epsilon(x)$.)

Axiom

Trivial.

Weakening

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash A : B}{\text{induction}}}{\langle\!|\Gamma|\!\rangle \vdash \langle\!|A|\!\rangle : \langle\!|B|\!\rangle} \qquad \dfrac{\dfrac{\Gamma \vdash C : s^n}{\text{induction}}}{\dfrac{\langle\!|\Gamma|\!\rangle \vdash \langle\!|C|\!\rangle : s}{\langle\!|\Gamma|\!\rangle, \langle\!|x_i : C|\!\rangle \text{ legal}}\ \text{Lem. 12}_{\text{iii}}}}{\langle\!|\Gamma|\!\rangle, \langle\!|x_i : C|\!\rangle \vdash \langle\!|A|\!\rangle : \langle\!|B|\!\rangle}\ \text{Thinning}$$

Application **(Rel-Elim is similar)**

$$\text{by def.}\ \dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash F : (\forall \bar{x} : \bar{A}.\, B)}{\text{induction}}}{\langle\!|\Gamma|\!\rangle \vdash \langle\!|F|\!\rangle : (\forall \langle\!|\bar{x} : \bar{A}|\!\rangle.\, \langle\!|B|\!\rangle)}}{\langle\!|\Gamma|\!\rangle \vdash \langle\!|F|\!\rangle : (\forall \{x^\pi_{ji} : \dots\}.\, \langle\!|B|\!\rangle)} \qquad \dfrac{\dfrac{\dfrac{\Gamma \vdash \bar{a} : \bar{A}}{\Gamma \vdash a_i : A_i\bullet(\bar{x} /\!\!/ i)}\ \text{by def.}}{\text{induction}}}{\langle\!|\Gamma|\!\rangle \vdash \langle\!|\, [\![a_i]\!]^{\|j\|} \ddagger^\pi \,|\!\rangle : \langle\!|\, (a_i \in [\![A_i\bullet(\bar{x} /\!\!/ i)]\!]^{\|j\|}) \ddagger^\pi \,|\!\rangle}}{\dfrac{\langle\!|\Gamma|\!\rangle \vdash \langle\!|F|\!\rangle \{\langle\!|\, [\![a_i]\!]^{\|j\|} \ddagger^\pi \,|\!\rangle \mid \dots\} : \langle\!|B|\!\rangle [\langle\!|\, [\![a_i]\!]^{\|j\|} \ddagger^\pi \,|\!\rangle / x^\pi_{ji}, \dots]}{\langle\!|\Gamma|\!\rangle \vdash \langle\!|F\,\bar{a}|\!\rangle : \langle\!|B[\bar{a}/\bar{x}]|\!\rangle}\ \text{by def., Lem. 15}}\ \text{(many-)App.}$$

Abstraction **(Rel-Intro is similar)**

$$\text{by def.}\ \dfrac{\dfrac{\dfrac{\dfrac{\Gamma, \bar{x} : \bar{A} \vdash b : B}{\text{induction}}}{\langle\!|\Gamma|\!\rangle, \langle\!|\bar{x} : \bar{A}|\!\rangle \vdash \langle\!|b|\!\rangle : \langle\!|B|\!\rangle} \quad \vdots}{\langle\!|\Gamma|\!\rangle \vdash (\lambda \langle\!|\bar{x} : \bar{A}|\!\rangle.\, \langle\!|b|\!\rangle : (\forall \langle\!|\bar{x} : \bar{A}|\!\rangle.\, \langle\!|B|\!\rangle}}{\langle\!|\Gamma|\!\rangle \vdash \langle\!|\lambda \bar{x} : \bar{A}.\, b|\!\rangle : \langle\!|\forall \bar{x} : \bar{A}.\, B|\!\rangle}\ \text{(many-)Abs.}$$

PRODUCT **(REL-FORM is similar)**

$$\text{by def.}\ \frac{\dfrac{\dfrac{\Gamma \vdash \bar{A} : s_1^m}{\Gamma, \ldots \vdash A_i{\bullet}(\bar{x} /\!\!/ i) : s_1^m}}{\text{induction}}}{\dfrac{\langle\!|\Gamma|\!\rangle, \ldots \vdash \langle\!|(x_i \in [\![A_i{\bullet}(\bar{x} /\!\!/ i)]\!]^{\|j\|})\ddagger^{\pi}|\!\rangle : s_1}{\langle\!|\Gamma|\!\rangle \vdash (\forall\{x_{ji}^{\pi} : \langle\!|(x_i \in [\![A_i{\bullet}(\bar{x} /\!\!/ i)]\!]^{\|j\|})\ddagger^{\pi}|\!\rangle \mid \ldots\}.\langle\!|B|\!\rangle) : s_3}\ \text{by def.}}\ \text{(many-)PROD.}\quad \frac{\dfrac{\dfrac{\Gamma, \bar{x} : \bar{A} \vdash B : s_2^n}{\text{induction}}}{\langle\!|\Gamma|\!\rangle, \langle\!|\bar{x} : \bar{A}|\!\rangle \vdash \langle\!|B|\!\rangle : s_2}}{\langle\!|\Gamma|\!\rangle \vdash \langle\!|\forall \bar{x} : \bar{A}.\, B|\!\rangle : s_3}$$

CONVERSION

$$\frac{\dfrac{\dfrac{\Gamma \vdash A : B}{\text{induction}}}{\langle\!|\Gamma|\!\rangle \vdash \langle\!|A|\!\rangle : \langle\!|B|\!\rangle} \qquad \dfrac{\dfrac{\Gamma \vdash B' : s^n}{\text{induction}}}{\langle\!|\Gamma|\!\rangle \vdash \langle\!|B'|\!\rangle : s} \qquad \dfrac{B =_\beta B'}{\dfrac{\text{Thm. 3, Lem. 16}}{\langle\!|B|\!\rangle =_\beta \langle\!|B'|\!\rangle}}}{\langle\!|\Gamma|\!\rangle \vdash \langle\!|A|\!\rangle : \langle\!|B'|\!\rangle}\ \text{CONV.}$$

START, PARAM, EXCHANGE

$$\frac{\dfrac{\dfrac{\dfrac{\Gamma \vdash A : s^m}{\text{induction}}}{\langle\!|\Gamma|\!\rangle, \langle\!|x_i : A|\!\rangle \text{ legal}}}{\langle\!|\Gamma|\!\rangle, \langle\!|x_i : A|\!\rangle \vdash x_{ji}^{\pi} : \langle\!|(x_i \in [\![A]\!]^n)\ddagger^{\pi}|\!\rangle}\ \text{Thinning, START}}{\langle\!|\Gamma|\!\rangle, \langle\!|x_i : A|\!\rangle \vdash \langle\!|\,[\![x_i]\!]^n\dagger^{\pi}|\!\rangle : \langle\!|(x_i \in [\![A]\!]^n)\ddagger^{\pi}|\!\rangle}\ \text{by def.}$$