Towards cubical type theory

Thorsten Altenkirch[1] **and Ambrus Kaposi**[1]

1 **University of Nottingham**
Nottingham, United Kingdom
{txa,auk}@cs.nott.ac.uk

Abstract

Our goal is to develop a type theory which provides a computational explanation of univalence. To achieve this goal we define a type theory with internal parametricity. The main idea is providing a syntax for the metatheoretic proof that every term is parametric, that is, its interpretations in related contexts are related. The function, which maps a term to its parametricity proof cannot be typed, because the domain and codomain live in different contexts. To solve this problem, we introduce a substitution from a context to the extended context of related elements. This work is a variant of Bernardy-Moulin's work on internal parametricity [5] but uses dimension names instead of permutations and is defined as an explicit substitution calculus.

We define an operational semantics for the theory.

The parametricity relation for the universe is just any relation. If we replace this by equivalence, our theory admits univalence. We discuss how this modification could be achieved.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases homotopy type theory, parametricity, univalence

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

Homotopy Type Theory [9] introduces the *univalence axiom*, which basically identifies propositional equality with equivalence of types[1]. We can understand the *univalence axiom* as a powerful extensionality principle which allows us to replace one type by another which has the same behaviour. It also entails functional extensionality.

The motivation for our work is that the current formulations of Homotopy Type Theory (as the one given in the appendix of [9]) lack a computational understanding of univalence. With some simplification, we add the axiom

$$\mathsf{univ} : A \simeq B \to A =_{\mathsf{U}} B$$

saying that an equality in the universe is provided given an equivalence between the types A and B. Here, equality is defined as an inductive family with the constructor refl and eliminator J. The axiom univ adds another constructor without providing an elimination rule for this constructor resulting in stuck terms. For example, we can define the equivalence $(\mathsf{not}, \dots)$ between Bool and Bool and thus $\mathsf{univ}\,(\mathsf{not}, \dots) : \mathsf{Bool} =_{\mathsf{U}} \mathsf{Bool}$, and then using it,

[1] Equivalence of types is a proof-relevant refinement of isomorphism. Naively it is sufficient to think of isomorphism since this notion is logically equivalent (but not isomorphic as a type).

Conference title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–19

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

define a boolean b:

$$\mathsf{coe} : (A =_{\mathsf{U}} B) \to A \to B$$
$$\mathsf{coe}\,(\mathsf{refl}\,A)\,a :\equiv a$$
$$b : \mathsf{Bool}$$
$$b :\equiv \mathsf{coe}\,(\mathsf{univ}\,(\mathsf{not}, \dots))\,\mathsf{true}$$

We know that b should be false as we coerce along not, but coe was only defined for the case refl[2] so the term b does not compute further. One may think that it would be sufficient to add the equation $\mathsf{coe}\,(\mathsf{univ}\,(f \dots)) \equiv f$ but this is not sufficient. Another example is obtained by using the ap (apply path) function for $P : \mathsf{U} \to \mathsf{U}$:

$$\mathsf{ap}_P : A =_{\mathsf{U}} B \to P\,A =_{\mathsf{U}} P\,B$$
$$\mathsf{ap}_P\,(\mathsf{refl}\,A) \equiv \mathsf{refl}\,(P\,A)$$
$$x : P\,\mathsf{Bool} =_{\mathsf{U}} P\,\mathsf{Bool}$$
$$x :\equiv \mathsf{ap}_P\,(\mathsf{univ}\,(\mathsf{not}, \dots))$$

To reduce this term we need to know how to transport an equivalence along an arbitrary type.

We observe that the problem is caused by viewing equality as an inductive type with only one constructor. Alternatively, we may want to exploit the characterisation of equalities for each type: equality of pairs is a pair of equality, equality of function is given by a function and equality of types is given by equivalence. That is we want to define equality as a logical relation as known from parametricity ([10, 8, 3]) and we want to exploit the fact that every term preserve logical relations to define ap, which is reflected in the fact that such a proof is available within the system.

A theory with internal parametricity was defined by Bernardy and Moulin [5] and given a nominal syntax by the same authors in [4]. This work can be seen as a reformulation of these as an explicit substitution calculus using dimension names instead of permutations. A difference from the latter work is that the dimensions of a term are implicit, we don't need to decorate typing judgments with the list of dimensions they are using.

Another line of research for providing computational meaning of univalence is defining models in a constructive metatheory. The cubical set model of type theory [6] is a presheaf model which is very close to our type theory defined below. The full cubical set model extends the presheaf model with Kan operations, this corresponds to the step of replacing the definition of "relatedness" for the universe by equivalence. We call this approach *cubical type theory* in reference to cubical sets and to the higher dimensional cubes which show up when we unfold our definition of equality.

Our contributions are the following:

- We show the problems arising when trying to define a syntax for parametricity naively (section 2).
- We define the syntax for the theory with internal parametricity by starting with a (standard) explicit substitution calculus without function types or equality, and then list the additional rules with explanation (section 3).
- We give a big-step operational semantics in the style of [7] (section 4). We conjecture that it is sound and complete.

[2] coe was defined by pattern matching, which can be seen as a usage of the eliminator J in this case.

- We discuss the necessary modifications needed to replace the relation for the universe with equivalence (section 5).

2 A naive explanation of parametricity

We start with a naive explanation of our basic ideas - the syntax is defined in more detail in section 3.1 extended with the usual rules for function space.

2.1 External parametricity

Parametricity says that logically related interpretations of a term are logically related. More precisely, given $\Gamma \vdash t : A$, $\rho, \rho' : \cdot \Rightarrow \Gamma$ and a witness that ρ and ρ' are related in the logical relation generated by Γ, then $t[\rho]$ and $t[\rho']$ will be related in the logical relation generated by A. We formalize this by the following rule:

$$\frac{\Gamma \vdash t : A}{\Gamma^i \vdash t^i : A^i\, t[\mathsf{0}_{i\Gamma}]\, t[\mathsf{1}_{i\Gamma}]}$$

Γ^i contains two copies of Γ (ρ and ρ' above) and a logical relation between them. A^i will be the logical relation generated by A. It relates the two interpretations of the term t, in the two different copies of Γ, which are projected out by the substitutions $\mathsf{0}_{i\Gamma}$, $\mathsf{1}_{i\Gamma}$. t^i will be the witness of this relation.

The above instance of parametricity used the name i. Each instance of parametricity will use a new dimension name. We denote these names by i, j, k. They are assumed to be fresh in every rule. The context Γ^i can be viewed as Γ with an added dimension named i.

The relation generated by a type A has the following specification:

$$\frac{\Gamma \vdash}{\Gamma^i \vdash} \qquad \frac{\Gamma \vdash A : \mathsf{U}}{\Gamma^i \vdash A^i : A[\mathsf{0}_{i\Gamma}] \to A[\mathsf{1}_{i\Gamma}] \to \mathsf{U}} \qquad \mathsf{0}_{i\Gamma}, \mathsf{1}_{i\Gamma} : \Gamma^i \Rightarrow \Gamma$$

The operation $-^i$ on contexts duplicates the context, and adds witnesses that the two new subcontexts are pointwise related.

$$\cdot^i \equiv \cdot$$
$$(\Gamma.x : A)^i \equiv \Gamma^i.x_{i0} : A[\mathsf{0}_{i\Gamma}].x_{i1} : A[\mathsf{1}_{i\Gamma}].x_{i2} : A^i\, x_{i0}\, x_{i1}$$

The substitutions $\mathsf{0}$ and $\mathsf{1}$ project out the corresponding components ($b = \mathsf{0}, \mathsf{1}$) while losing the dimension i.

$$b_{i\cdot} \equiv () : \cdot \Rightarrow \cdot$$
$$b_{i(\Gamma.x:A)} \equiv (b_{i\Gamma}, x \mapsto x_{ib}) : (\Gamma.x : A)^i \Rightarrow \Gamma.x : A$$

The relation A^i is generated by induction on A: for function types, it says that related inputs are mapped to related outputs, for type variables, it is relation space. As types are just terms, we define t^i for every term t uniformly, including the cases when it is a type:

$$\begin{aligned}
(\Pi(x : A).B)^i &\equiv \lambda f_{i0}\, f_{i1}.\Pi(x_{i0} : A[\mathsf{0}_i], x_{i1} : A[\mathsf{1}_i], x_{i2} : A^i\, x_{i0}\, x_{i1}).B^i\, (f_{i0}\, x_{i0})\, (f_{i1}\, x_{i1}) \\
\mathsf{U}^i &\equiv \lambda X_{i0}\, X_{i1}.X_{i0} \to X_{i1} \to \mathsf{U} \\
(\lambda x.t)^i &\equiv \lambda x_{i0}\, x_{i1}\, x_{i2}.t^i \\
(f\, u)^i &\equiv f^i\, u[\mathsf{0}_i]\, u[\mathsf{1}_i]\, u^i \\
x^i &\equiv x_{i2} \\
(t[\rho])^i &\equiv t^i[\rho^i]
\end{aligned}$$

Note that U^i validates the rule $\mathsf{U}^i : \mathsf{U}^i \,\mathsf{U}\, \mathsf{U}$ by reducing its type. Given a $\rho : \Delta \Rightarrow \Gamma$, $\rho^i : \Delta^i \Rightarrow \Gamma^i$ can be defined pointwise on a substitution, we omit these rules for now.

The above rules added external parametricity to the theory. For example, using the new rules, we can derive that a particular inhabitant of the type $\Pi(A : \mathsf{U}).A \to A$ is the identity function. We denote the Leibniz equality of the type A by Id_A and its constructor by refl.

$$\frac{f : \Pi(A : \mathsf{U}).A \to A}{f^i : \Pi \begin{pmatrix} A_{i0} : \mathsf{U}, A_{i1} : \mathsf{U}, A_{i2} : A_{i0} \to A_{i1} \to \mathsf{U}, \\ x_{i0} : A_{i0}, x_{i1} : A_{i1}, x_{i2} : A_{i2}\, x_{i0}\, x_{i1} \end{pmatrix} . A_{i2}\, (f\, A_{i0}\, x_{i0})\, (f\, A_{i1}\, x_{i1})}$$

and hence

$$\lambda A : \mathsf{U}\, y : A. f^i\, A\, A\, (\lambda x_{i0}\, x_{i1}.\mathsf{Id}_A\, x_{i0}\, y)\, y\, y\, (\mathsf{refl}\, y) : \Pi(A : \mathsf{U}, y : A).\mathsf{Id}_A\, (f\, A\, y)\, y$$

The operational semantics for the above theory is clear, we can always eliminate the new constructs $-^i$, $\mathsf{0}_i$, $\mathsf{1}_i$.

We call this theory *external* because even though we can show for every instance of $f : \Pi(A : \mathsf{U}).A \to A$ that it behaves like the identity but we cannot show this *internally* for an assumption of the form $f : \Pi(A : \mathsf{U}).A \to A$.

2.2 Internal parametricity

To internally derive that every function of type $\Pi(A : \mathsf{U}).A \to A$ is the identity, we would need a term t expressing parametricity for f assuming it in the context (and not as a closed term):

$$f : \Pi(A : \mathsf{U}).A \to A \vdash t : \Pi(A_{i0}, A_{i1} : \mathsf{U}, A_{i2} : A_{i0} \to A_{i1} \to \mathsf{U})\,. \\ \Pi(x_{i0} : A_{i0}, x_{i1} : A_{i1}, x_{i2} : A_{i2}\, x_{i0}\, x_{i1})\,. A_{i2}\, (f\, A_{i0}\, x_{i0})\, (f\, A_{i1}\, x_{i1}).$$

We could choose t to be f^i, but that lives in the context $(f : \Pi(A : \mathsf{U}).A \to A)^i$.

This motivates the definition of a substitution that takes us into the $-^i$-d context. The rules are as follows:

$$\frac{\Gamma \vdash}{\mathsf{R}_{i\Gamma} : \Gamma \Rightarrow \Gamma^i} \qquad \mathsf{R}_{i\cdot} \equiv () \qquad \mathsf{R}_{i(\Gamma.x:A)} \equiv (\mathsf{R}_{i\Gamma}, x_{i0} \mapsto x, x_{i1} \mapsto x, x_{i2} \mapsto x^i[\mathsf{R}_{i(\Gamma.x:A)}])$$

We also add the following computation rule:

$$\frac{\rho : \Delta \Rightarrow \Gamma}{\mathsf{R}_{i\Gamma}\rho \equiv \rho^i \mathsf{R}_{i\Delta}}$$

Note that $x^i[\mathsf{R}_i]$ is a new normal form (when x is a variable) as the computation rule doesn't give anything new.

By adding new term formers, we need to say how the $-^i$ operation acts on them, i.e. what is $(\mathsf{R}_{j\Gamma})^i : \Gamma^i \Rightarrow \Gamma^{ji}$. Note that $\mathsf{R}_{j\Gamma^i}$ has a different type, $\Gamma^i \Rightarrow \Gamma^{ij}$, and if we would equate $\mathsf{R}_{j\Gamma})^i$ and $\mathsf{R}_{j\Gamma^i}$ and their types, our theory would become inconsistent. For example, composing the substitution

$$\rho \equiv (x_{i0} \mapsto a, x_{i1} \mapsto b, x_{i2} \mapsto c) : \cdot \Rightarrow (x : A)^i$$

with the above two substitutions, we get the following results:

$$\mathsf{R}_{j(x:A)^i}\rho \equiv (a, a, a^j[\mathsf{R}_j], b, b, b^j[\mathsf{R}_j], c \quad , c \quad , c^j[\mathsf{R}_j] \quad) : \cdot \Rightarrow (x : A)^{ij}$$
$$(\mathsf{R}_{j(x:A)})^i\rho \equiv (a, b, c \quad , a, b, c \quad , a^j[\mathsf{R}_j], b^j[\mathsf{R}_j], x_{j2i2}[(\mathsf{R}_j)^i\rho]) : \cdot \Rightarrow (x : A)^{ji}$$

If we equate these two, as the elements are in different order, we can prove $\mathsf{true} \equiv \mathsf{false}$ by setting A to Bool, a to true and b to false.

However, the x_{i1j0} element of the first substitution is the same as the x_{j0i1} element of the second substitution etc. This suggests the view of a context $(x : A)^{ij}$ as a type for each $i = 0, 1, 2$ and $j = 0, 1, 2$ regardless of the order; i and j are the dimension names, and given a coordinate for each dimension, we get a type. We will follow this path in the next section. We will also need to adjust Π types so that their dimensions are explicit: eg. given a function $f : \Pi(x : A).B$, we have

$$(f^i)^j : \Pi(x : A)^{ij}.(B^i\,(f[\mathsf{0}_i]\,x_{i0})\,(f[\mathsf{1}_i]\,x_{i1}))^j\,(f^i[\mathsf{0}_j]\,x_{i0j0}\,x_{i1j0}\,x_{i2j0})\,(f^i[\mathsf{1}_j]\,x_{i0j1}\,x_{i1j1}\,x_{i2j1})$$

and the arguments of f^{ij} will need to be given for each coordinate and not just as a sequence.

In this section, we added internal parametricity to our theory, so know we can use $f^i[\mathsf{R}_i]$ as t in the above example:

$$\begin{aligned}&\lambda f\,A\,y.f^i[\mathsf{R}_{i(f:\Pi(A:\mathsf{U}).A\to A)}]\,A\,A\,(\lambda x_0\,x_1.\mathsf{Id}_A\,x_0\,y)\,y\,y\,(\mathsf{refl}\,y)\\ &: \Pi\big(f : \Pi(A : \mathsf{U}).A \to A, A : \mathsf{U}, x : A\big).\mathsf{Id}_A\,(f\,A\,x)\,x\end{aligned}$$

However, we lack some computation rules as exemplified by $(\mathsf{R}_{j\Gamma})^i$.

3 A syntax for internal parametricity

This section is a listing of derivation rules for our theory with internal parametricity along with explanations. All of the rules listed are part of the calculus, we only write "defined by induction" or "specification" and "implementation" for intuition. We can view the previous section about the naive implementation as a specification or inspiration for this section, but we won't make use of any of its rules, we are defining a new theory.

3.1 Type theory with explicit substitutions

We start with an explicit substitution calculus with variable names and universes à la Russell. We assume fresh names in every rule. α-equivalence and weakening are implicit. We use $\mathsf{U} : \mathsf{U}$ for presentation purposes, but mean $\mathsf{U}_i : \mathsf{U}_j$ only if $i < j$ officially.

Notations:

Γ, Δ, Θ	contexts
x, y, z, X	variables
t, u, v, f	terms
A, B, C	types
ρ, σ, ν	substitutions

Judgment types:

$\Gamma \vdash$	Γ is a valid context
$\Gamma \vdash t : A$	t is a term of type A in context Γ
$\rho : \Delta \Rightarrow \Gamma$	a substitution from Δ to Γ
$\Gamma \equiv \Gamma'$	definitional equality of contexts
$\Gamma \vdash t \equiv t' : A$	definitional equality of terms
$\Gamma \vdash \rho \equiv \rho' : \Delta \Rightarrow \Gamma$	definitional equality of substitutions

Rules:

$$\frac{}{\cdot \vdash} \qquad \frac{\Gamma \vdash \quad \Gamma \vdash A : \mathsf{U}}{\Gamma.x : A \vdash}$$

$$\frac{\Gamma \vdash}{() : \Gamma \Rightarrow \cdot} \qquad \frac{\Gamma \vdash A : \mathsf{U} \quad \rho : \Delta \Rightarrow \Gamma \quad \Delta \vdash t : A[\rho]}{(\rho, x \mapsto t) : \Delta \Rightarrow \Gamma.x : A} \qquad \frac{\Gamma \vdash}{\mathsf{id}_\Gamma : \Gamma \Rightarrow \Gamma}$$

$$\frac{\rho : \Delta \Rightarrow \Gamma \quad \sigma : \Theta \Rightarrow \Delta}{\rho\sigma : \Theta \Rightarrow \Gamma} \qquad \frac{\Delta \vdash A : \mathsf{U} \quad \rho : \Delta \Rightarrow \Gamma}{\rho : \Delta.x : A \Rightarrow \Gamma}$$

$$\mathsf{id}\rho \equiv \rho \equiv \rho\mathsf{id} \qquad \nu(\rho\sigma) \equiv (\nu\rho)\sigma \qquad (\rho, x \mapsto t)\sigma \equiv (\rho\sigma, x \mapsto t[\sigma])$$

$$(\underbrace{\mathsf{id}_\Gamma}_{:\Gamma.x:A \Rightarrow \Gamma}, x \mapsto x) \equiv \mathsf{id}_{\Gamma.x:A}$$

$$\frac{\Gamma \vdash A : \mathsf{U} \quad \Gamma \vdash t : B}{\Gamma.x : A \vdash t : B} \qquad \frac{\Gamma \vdash t : A \quad \rho : \Delta \to \Gamma}{\Delta \vdash t[\rho] : A[\rho]} \qquad t[\mathsf{id}] \equiv t \qquad t[\rho][\sigma] \equiv t[\rho\sigma]$$

$$\frac{\Gamma \vdash A : \mathsf{U}}{\Gamma.x : A \vdash x : A} \qquad \frac{\Gamma \vdash}{\Gamma \vdash \mathsf{U} : \mathsf{U}} \qquad \mathsf{U}[\rho] \equiv \mathsf{U}$$

We assume all congruence rules and reflexivity, symmetry, transitivity of $\equiv$. We usually omit the starting $\cdot$ from contexts and $()$ from substitutions. While we present the system using named variables for readability we assume that terms are identified upto α-conversion and that name capture is always avoided by appropriate renaming.

3.2 Telescopes

We add new judgment types:

$\Gamma \vdash \Omega$	Ω is a telescope context in context Γ
$\Gamma \vdash \omega : \Omega$	ω is a telescope substitution of type Ω in context Γ
$\Gamma \vdash \Omega \equiv \Omega'$	definitional equality of telescope contexts
$\Gamma \vdash \omega \equiv \omega' : \Omega$	definitional equality of telescope substitutions

Explanation:

Just as	$\Gamma \vdash t : A$	can be viewed as	$(\mathsf{id}_\Gamma, x \mapsto t) : \Gamma \Rightarrow \Gamma.x : A$,
	$\Gamma \vdash \Omega$	can be viewed as	$\Gamma +\!\!+ \Omega \vdash$
and	$\Gamma \vdash \omega : \Omega$	can be viewed as	$(\mathsf{id}_\Gamma +\!\!+ \omega) : \Gamma \Rightarrow \Gamma.\Omega$.

Telescope contexts are generalisations of types into lists of types which might depend on each other. They can also be thought of as named iterated Σ-types.

$$\frac{\Gamma \vdash}{\Gamma \vdash \cdot} \qquad \frac{\Gamma \vdash \quad \Gamma \vdash \Omega \quad \Gamma.\Omega \vdash A : \mathsf{U}}{\Gamma \vdash \Omega.x : A}$$

We explain how to extend contexts with telescope contexts:

$$\frac{\Gamma \vdash \Omega}{\Gamma +\!\!+ \Omega \vdash} \qquad \Gamma +\!\!+ \cdot \equiv \Gamma \qquad \Gamma +\!\!+ (\Omega.x : A) \equiv (\Gamma +\!\!+ \Omega).x : A$$

Substitution of telescope contexts is pointwise:

$$\frac{\Gamma \vdash \Omega \quad \rho : \Delta \Rightarrow \Gamma}{\Delta \vdash \Omega[\rho]} \qquad \cdot[\rho] \equiv \cdot \qquad (\Omega.x : A)[\rho] \equiv \Omega[\rho].x : A[\rho]$$

Telescope substitutions are generalisations of terms into lists of terms.

$$\frac{\Gamma \vdash}{\Gamma \vdash () : \cdot} \qquad \frac{\Gamma \vdash \omega : \Omega \quad \Gamma.\Omega \vdash A : \mathsf{U} \quad \Gamma \vdash t : A[(\mathsf{id}_\Gamma, \omega)]}{\Gamma \vdash (\omega, x \mapsto t) : \Omega.x : A}$$

We explain how to extend normal substitutions with telescope substitutions:

$$\frac{\rho : \Delta \Rightarrow \Gamma \quad \Delta \vdash \omega : \Omega[\rho]}{\rho +\!\!+ \omega : \Delta \Rightarrow \Gamma +\!\!+ \Omega} \qquad \rho +\!\!+ () \equiv \rho \qquad \rho +\!\!+ (\omega, x \mapsto t) \equiv (\rho +\!\!+ \omega, x \mapsto t)$$

Substitution of telescope substitutions is pointwise:

$$\frac{\Gamma \vdash \omega : \Omega \quad \rho : \Delta \Rightarrow \Gamma}{\Delta \vdash \omega[\rho] : \Omega[\rho]} \qquad ()[\rho] \equiv () \qquad (\omega, x \mapsto t)[\rho] \equiv (\omega[\rho], x \mapsto t[\rho])$$

Extending telescope contexts with telescope contexts and telescope substitutions with telescope substitutions is done in the obvious way. Specification:

$$\frac{\Gamma \vdash \Omega \quad \Gamma +\!\!+ \Omega \vdash \Omega'}{\Gamma \vdash \Omega +\!\!+ \Omega'} \qquad \Omega +\!\!+ \cdot \equiv \Omega \qquad \Omega +\!\!+ (\Omega'.x : A) \equiv \Omega +\!\!+ \Omega'.x : A$$

$$\frac{\Gamma \vdash \omega : \Omega \quad \Gamma \vdash \omega' : \Omega'[\omega]}{\Gamma \vdash \omega +\!\!+ \omega' : \Omega +\!\!+ \Omega'} \qquad \omega +\!\!+ () \equiv \omega \qquad \omega +\!\!+ (\omega', x \mapsto t) \equiv (\omega +\!\!+ \omega', x \mapsto t)$$

The pr telescope substitution generalises the projection $\Gamma.x : A \vdash x : A$.

$$\frac{\Gamma \vdash \Omega}{\Gamma +\!\!+ \Omega \vdash \mathsf{pr}_\Omega : \Omega} \qquad \Gamma +\!\!+ \cdot \vdash \mathsf{pr}_\cdot \equiv () : \cdot \qquad \Gamma +\!\!+ (\Omega.x : A) \vdash \mathsf{pr}_{\Omega.x:A} \equiv (\mathsf{pr}_\Omega, x \mapsto x) : (\Omega.x : A)$$

Note that we haven't added new types in the universe, telescopes live outside the world of types.

3.3 Function space

We would like to have $\Gamma^{ij} \equiv \Gamma^{ji}$ to be able to compute $(\mathsf{R}_{j\Gamma})^i$ as described in section 2.2. This forces us to provide a new type former for $\Pi(x : A)^{ij}.B$ instead of using the iterated $\Pi(x_{i0j0} : A[\mathsf{0}_i\mathsf{0}j]).\Pi(x_{i0j1} : A[\mathsf{0}_i\mathsf{1}_j]).\ldots.B$. When applying arguments to such a function, we need to supply all of them at the same time, because the order of arguments won't matter, only their dimension indices.

For this reason, we add functions where the domain can be a telescope context. However, we only allow special telescope contexts, ones which arise from applying the lifting operator zero or more times. We will denote such a context $(x : A)^I$, where I is a set of dimensions which might be empty.

Also, to express the type of the relation $A^i : A[\mathsf{0}_i] \to A[\mathsf{1}_i] \to U$, we need functions with incomplete cube arguments, $\{x : A\}_I$ will denote $(x : A)^I$ without the last element, so $A^i : \Pi\{x : A\}_I.\mathsf{U}$. We will add rules to compute $(x : A)^I$ and $\{x : A\}_I$ in the next section.

Our function space also needs to be stable under substitution, for example, given a type $(X : \mathsf{U})^i \vdash \Pi(x : X)^i.B : \mathsf{U}$ and a $\rho : \cdot \Rightarrow (X : \mathsf{U})^i$, $(\Pi(x : X)^i.B)[\rho]$ doesn't have the form of $\Pi(x : A)^i.B$ for some A anymore. This is why the function domain is decorated with a telescope substitution providing the types. The formation rules:

$$\frac{\Gamma \vdash \xi : (X : \mathsf{U})^I \quad \Gamma +\!\!+ (x : X)^I[\xi] \vdash B : \mathsf{U}}{\Gamma \vdash \Pi(x : X)^I[\xi].B : \mathsf{U}} \qquad \frac{\Gamma \vdash \xi : \{X : \mathsf{U}\}_{Ii}}{\Gamma \vdash \Pi\{x : X\}_{Ii}[\xi].\mathsf{U} : \mathsf{U}}$$

These are just the usual rules for telescope functions, restricted on the domain (and for relations, in the codomain too). Note that the domain for relation space can't be zero-dimensional. Also, we haven't defined substituting a telescope context with a telescope substitution, so we mean $(x : X)^I[\mathsf{id} +\!\!+ \xi]$ when we write $(x : X)^I[\xi]$.

Most rules have the same shape for functions and relations, for these, we introduce the notation $(\!|x : A|\!)_I$ which can mean $(x : A)^I$ or $\{x : A\}_I$. We use the notation $\mathsf{app}^I(f, \omega)$ function application, $\mathsf{app}_I(R, \omega)$ for relation application and $\mathsf{app}_I(f, \omega)$ can mean be both.

$$\frac{\Gamma +\!\!+ (\!|x : X|\!)_I[\xi] \vdash t : B}{\Gamma \vdash \lambda(\!|x : X|\!)_I[\xi].t : \Pi(\!|x : X|\!)_I[\xi].B} \qquad \frac{\Gamma \vdash f : \Pi(\!|x : X|\!)_I[\xi].B \quad \Gamma \vdash \omega : (\!|y : X|\!)_I[\xi]}{\Gamma \vdash \mathsf{app}_I(f, \omega) : B[\mathsf{id} +\!\!+ \omega]}$$

The computation rules include η:

$$\begin{aligned}
\mathsf{app}_I(\lambda(\!|x|\!)_I.t, \omega) &\equiv t[\mathsf{id} +\!\!+ \omega] \\
f &\equiv \lambda(\!|x : X|\!)_I[\xi].\mathsf{app}_I(f, \mathsf{pr}_{\{x:X\}I[\xi]}) \\
(\Pi(\!|x : X|\!)_I[\xi].B)[\sigma] &\equiv \Pi(\!|x : X|\!)_I[\xi[\sigma]].B[\sigma +\!\!+ \mathsf{pr}_{\{x:X\}I[\xi[\sigma]]}] \\
(\lambda(\!|x : X|\!)_I[\xi].t)[\sigma] &\equiv \lambda(\!|x : X|\!)_I[\xi[\sigma]].t[\sigma +\!\!+ \mathsf{pr}_{\{x:X\}I[\xi[\sigma]]}] \\
\mathsf{app}_I(f, \omega)[\sigma] &\equiv \mathsf{app}_I(f[\sigma], \omega[\sigma])
\end{aligned}$$

If the telescope substitution ξ is the projection pr, we omit it, eg. we write $\Pi(x : X).B$ for $\Pi(x : X)^I[\mathsf{pr}].B$. Also, we write $\Pi(x : A)^I.B$ for $\Pi(x : X)^I[(X \mapsto A)^I]$ and $\Pi\{x : A\}_I.B$ for $\Pi\{x : X\}_I[\{X \mapsto A\}_I]$.

Note that in the case when I is empty, $\Pi(x : A)^I.B$ becomes the usual function space, we denote it as $\Pi(x : A).B$.

3.4 The operations $(-)^i$ and $\{-\}_i$ on contexts and substitutions

We define $(-)^i$ and $\{-\}_i$ on contexts, telescope contexts, substitutions, telescope substitutions. The definitions for contexts are the same as in the naive version.

Specification:

$$\frac{\Gamma \vdash}{(\Gamma)^i \vdash} \qquad \frac{\rho : \Delta \Rightarrow \Gamma}{(\rho)^i : \Delta^i \Rightarrow \Gamma^i} \qquad \frac{\Gamma \vdash \Omega}{(\Gamma)^i \vdash \Omega^i} \qquad \frac{\Gamma \vdash \omega : \Omega}{(\Gamma)^i \vdash \omega^i : \Omega^i}$$

$$\frac{\Gamma.x : A \vdash}{\{\Gamma.x : A\}_i \vdash} \qquad \frac{\rho : \Delta \Rightarrow \Gamma.x : A}{\{\rho\}_i : \Delta^i \Rightarrow \{\Gamma.x : A\}_i} \qquad \frac{\Gamma \vdash \Omega.x : A}{\Gamma^i \vdash \{\Omega.x : A\}_i} \qquad \frac{\Gamma \vdash \omega : \Omega.x : A}{\Gamma^i \vdash \{\omega\}_i : \{\Omega.x : A\}_i}$$

Implementation:

$$
\begin{array}{lll}
(\cdot)^i & \equiv \cdot & \\
(\Gamma.x : A)^i & \equiv \Gamma^i +\!\!+ (x : A)^i & \\
()^i & \equiv () & : \Gamma^i \Rightarrow \cdot^i \\
(\rho, x \mapsto t)^i & \equiv (\rho)^i +\!\!+ (x \mapsto t)^i & : \Delta^i \Rightarrow (\Gamma.x : A)^i \\
\Gamma^i \vdash (\cdot)^i & \equiv \cdot & \\
\Gamma^i \vdash (\Omega.x : A)^i & \equiv \Omega^i.x_{i0} : A[\mathsf{0}_{i\Gamma}].x_{i1} : A[\mathsf{1}_{i\Gamma}].x_{i2} : \mathsf{app}_i(A^i, (x)_i) & \\
\Gamma^i \vdash ()^i & \equiv () & : \cdot^i \\
\Gamma^i \vdash (\omega, x \mapsto t)^i & \equiv (\omega^i, x_{i0} \mapsto t[\mathsf{0}_{i\Gamma}], x_{i1} \mapsto t[\mathsf{1}_{i\Gamma}], x_{i2} \mapsto t^i) & : (\Omega.x : A)^i \\
\{\Gamma.x : A\}_i & \equiv \Gamma^i +\!\!+ \{x : A\}_i & \\
\{\rho, x \mapsto t\}_i & \equiv \rho^i +\!\!+ \{x \mapsto t\}_i & : \Delta^i \Rightarrow \{\Omega.x : A\}_i \\
\Gamma^i \vdash \{\Omega.x : A\}_i & \equiv \Omega^i.x_{i0} : A[\mathsf{0}_{i\Gamma}].x_{i1} : A[\mathsf{1}_{i\Gamma}] & \\
\Gamma^i \vdash \{\omega, x \mapsto t\}_i & \equiv (\omega^i, x_{i0} \mapsto t[\mathsf{0}_{i\Gamma}], x_{i1} \mapsto t[\mathsf{1}_{i\Gamma}]) & : \{\Omega.x : A\}_i
\end{array}
$$

The operation $\{\text{–}\}_i$ does the same as $(\text{–})^i$ but omits about the very last element (because of this, it does not work on empty contexts or substitutions).

The substitutions $\mathsf{0}$ and $\mathsf{1}$ project out the corresponding components ($b = \mathsf{0}, \mathsf{1}$) while losing the dimension i.

$$
\begin{array}{ll}
b_{i\cdot} & \equiv () : \cdot \Rightarrow \cdot \\
b_{i(\Gamma.x:A)} & \equiv (b_{i\Gamma}, x \mapsto x_{ib}) : (\Gamma.x : A)^i \Rightarrow \Gamma.x : A
\end{array}
$$

We also add how $(\text{–})^i$ operates on special substitutions:

$$(\rho\sigma)^i \equiv \rho^i\sigma^i \qquad (\mathsf{id}_\Gamma)^i \equiv \mathsf{id}_{\Gamma^i}$$

$(\text{–})^I$ is an iterated version of $(\text{–})^i$. $\{\text{–}\}_{Ii}$ is just $(\text{–})^I$ composed by $\{\text{–}\}_i$. For contexts we express this by the following rules, for the substitutions etc. we have analogous rules.

$$\Gamma^\emptyset \equiv \Gamma \qquad \Gamma^{Ij} \equiv (\Gamma^I)^j \qquad \{\Gamma\}_{Ij} \equiv \{\Gamma^I\}_j$$

An element of a (telescope) context produced by multiple applications of $(\text{–})^i$ can be viewed as an $|I|$-dimensional cube. We depict elements of $(x : A)^I$ for the first few dimensions on figure 1.

3.5 $(\text{–})^i$ on terms

The operation which maps a term to the witness of parametricity is the same as in the naive version, but with the additional information added for the function space. we need to handle functions and relations separately because performing $(\text{–})^i$ on a full cube gives a full cube, but on incomplete cubes it does not even give an incomplete cube. We need to fill in the omitted two elements in the latter case. Note that $(x : A)^I$ has $3^{|I|}$ elements and $\{x : A\}_I$ has $3^{|I|} - 1$ elements.

Specification:

$$\frac{\Gamma \vdash t : A}{\Gamma^i \vdash t^i : \mathsf{app}_i(A^i, \{x \mapsto t\}_i)} \tag{1}$$

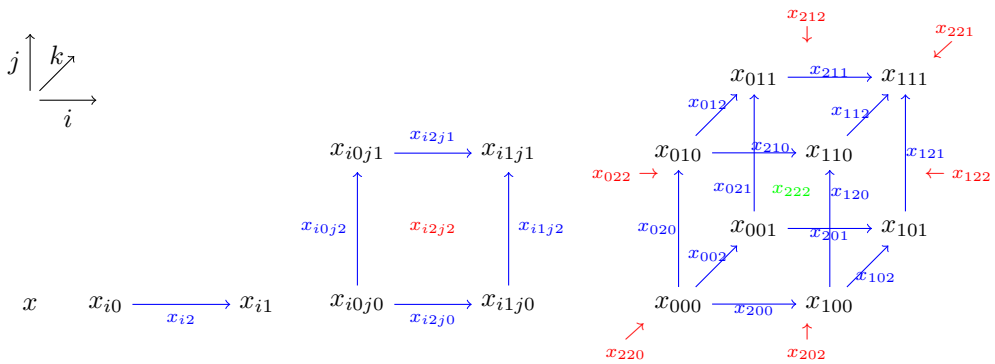

Figure 1 Elements of a context $(x : A)^I$ for different Is. In the last case, we didn't write the indices for readability, x_{abc} should be read as x_{iajbkc}. The directions of the dimensions are denoted in the upper left hand corner.

Implementation:

$$
\begin{aligned}
(t[\rho])^i &\equiv t^i[\rho^i] \\
x^i &\equiv x_{i2} \\
\mathsf{U}^i &\equiv \lambda\{X\}_i.\Pi\{x : X\}_i.\mathsf{U} \\
(\Pi(x : X)^I[\xi].B)^i &\equiv \lambda\{f\}_i.\Pi(x : X)^{Ii}[\xi^i].\mathsf{app}_i(B^i, \{y \mapsto \mathsf{app}^I(f, (x)^I)\}_i) \\
(\lambda(x)^I.t)^i &\equiv \lambda(x)^{Ii}.t^i \\
(\mathsf{app}^I(f, \omega))^i &\equiv \mathsf{app}^{Ii}(t^i, \omega^i) \\
(\Pi\{x : X\}_I[\xi].\mathsf{U})^i &\equiv \lambda\{y\}_i.\Pi\{x : X\}_{Ii}[\xi^i +\!\!+ \{X_{I2} \mapsto y\}_i].\mathsf{U} \\
(\lambda\{x\}_I.B)^i &\equiv \lambda\{x\}_{Ii}.\mathsf{app}_i(B^i, \{y \mapsto x_{I2}\}_i) \\
(\mathsf{app}_I(R, \omega))^i &\equiv \lambda\{y\}_i.\mathsf{app}_{Ij}(R^i, \omega^i +\!\!+ \{x_{I2} \mapsto y\}_i)
\end{aligned}
$$

x_{I2} is a notation for the variable x where all the dimensions have index 2, eg. x_{ijk2} is x_{i2j2k2}.

3.6 A definitional quotient

Now we can add rules equating $(\text{–})^{ij}$ and $(\text{–})^{ji}$.

$$\frac{\Gamma \vdash}{(\Gamma^i)^j \equiv (\Gamma^j)^i} \qquad \frac{\Gamma \vdash \Omega}{\Gamma^{ij} \vdash (\Omega^i)^j \equiv (\Omega^j)^i} \qquad \frac{\Gamma \vdash \omega : \Omega}{\Gamma^{ij} \vdash (\omega^i)^j \equiv (\omega^j)^i : \Omega^{ij}}$$

$$\frac{\rho : \Delta \Rightarrow \Gamma}{(\rho^i)^j \equiv (\rho^j)^i : \Delta^{ij} \Rightarrow \Gamma^{ij}} \qquad \frac{\Gamma \vdash t : A}{\Gamma^{ij} \vdash (t^i)^j \equiv (t^j)^i : \mathsf{app}_{ij}(A^{ij}, \{x \mapsto t\}_{ij})}$$

$$\frac{\Gamma \vdash}{\{\Gamma^i\}_j \equiv \{\Gamma^j\}_i} \qquad \frac{\Gamma \vdash \Omega}{\Gamma^{ij} \vdash \{\Omega^i\}_j \equiv \{\Omega^j\}_i} \qquad \frac{\Gamma \vdash \omega : \Omega}{\Gamma^{ij} \vdash \{\omega^i\}_j \equiv \{\omega^j\}_i : \{\Omega\}_{ij}}$$

$$\frac{\rho : \Delta \Rightarrow \Gamma}{\{\rho^i\}_j \equiv \{\rho^j\}_i : \Delta^{ij} \Rightarrow \{\Gamma\}_{ij}}$$

So we can treat the dimensions of a context, substitution etc. as a set and write them without parentheses.

Now we have $(b_{i\Gamma})^j \equiv b_{i\Gamma^j}$ for $b = 0, 1$.

3.7 Internalisation of parametricity

We add a substitution R that adds a dimension to a context.

$$\frac{\Gamma \vdash}{\mathsf{R}_{i\Gamma} : \Gamma \Rightarrow \Gamma^i} \qquad \frac{\Gamma \vdash t : A}{\Gamma \vdash t^{ⓘ} : \mathsf{app}_i(A^i[\mathsf{R}_{i\Gamma}], (x_{i0} \mapsto t, x_{i1} \mapsto t))}$$

$$\mathsf{R}_{i\cdot} \equiv () \qquad : \cdot \Rightarrow \cdot$$

$$\mathsf{R}_{i(\Gamma.x:A)} \equiv (\mathsf{R}_{i\Gamma}, x_{i0} \mapsto x, x_{i1} \mapsto x, x_{i2} \mapsto x^{ⓘ}) : (\Gamma.x : A) \Rightarrow (\Gamma.x : A)^i$$

$$t^{ⓘ} \equiv t^i[\mathsf{R}_{i\Gamma}] \qquad \frac{\rho : \Delta \to \Gamma}{\mathsf{R}_{i\Gamma}\rho \equiv \rho^i \mathsf{R}_{i\Delta}}$$

Note that we can derive $a^{ⓘ}[\rho] \equiv (a[\rho])^{ⓘ}$ from the above rules. Also, the computation rule doesn't give us anything new if a is a variable:

$$\Gamma.x : A \vdash x^{ⓘ} \equiv x^i[\mathsf{R}_{i(\Gamma.x:A)}] \equiv x_{i2}[\mathsf{R}_{i(\Gamma.x:A)}] \equiv x^{ⓘ}.$$

We finally add the rule describing how $(\text{–})^i$ works on the substitution R and how it interacts with $b = \mathsf{0}, \mathsf{1}$:

$$(\mathsf{R}_{j\Gamma})^i \equiv \mathsf{R}_{j\Gamma^i} \qquad b_{i\Gamma}\mathsf{R}_{i\Gamma} \equiv \mathsf{id}_\Gamma$$

The first rule needs the definitional quotient to typecheck.

With this rule, we defined a type theory with internal parametricity. In the next section we show how to normalise terms.

4 Operational semantics

In this section we describe a call by name operational semantics for the theory defined in section 3. We conjecture that the semantics defined below is terminating, and when viewed as a syntactic model construction, sound and complete (we refer to [1] for the meaning of these properties); decidability of definitional equality follows. The presentation is similar to that of [7].

We define values (weak-head normal forms), and show how to evaluate terms into values. Next we define normal forms, and show that by recursively applying evaluation, we can normalise terms. We describe the interesting details of evaluation and give the full formal definition in the appendix A.

In what follows, $\{\!|\text{–}|\!\}_I$ can mean both $(\text{–})^I$ and $\{\text{–}\}_I$. Note that in the latter case I is nonempty. We define the following inductive sets by listing their constructors:

t, A	$::= t[\rho] \mid x \mid \mathsf{U} \mid \Pi(x : X)^I[\omega].A \mid \Pi\{x : X\}_I[\omega].\mathsf{U}$	
	$\mid \lambda\{\!|x|\!\}_I.t \mid \mathsf{app}_I(t, \omega) \mid t^i \mid t^{ⓘ}$	terms
ρ	$::= \mathsf{id} \mid () \mid (\rho, x \mapsto t) \mid \rho\rho' \mid \rho +\!\!+ \omega \mid \mathsf{0}_i \mid \mathsf{1}_i \mid \mathsf{R}_i \mid \rho^i \mid \{\rho\}_i$	substitutions
ω	$::= () \mid (\omega, x \mapsto t) \mid \omega[\rho] \mid \omega +\!\!+ \omega' \mid \mathsf{pr}_\Omega \mid \omega^i \mid \{\omega\}_i$	telescope substitutions
x, f, X	$::= \ldots$	variables
v	$::= \mathsf{U} \mid (\Pi\{\!|x : X|\!\}_I[\omega].A)[\nu] \mid (\lambda\{\!|x|\!\}_I.t)[\nu] \mid n$	values (whnfs)
ν	$::= () \mid (\nu, x \mapsto g) \mid (\nu, x \mapsto t[\nu])$	environments
n	$::= g \mid \mathsf{app}_I(n, \psi)$	neutral values
g	$::= x \mid g^{ⓘ}$	generic values
ψ	$::= () \mid (\psi, x \mapsto g) \mid (\psi, x \mapsto t[\nu])$	telescope environments

We have the following judgment types for evaluation:

$t[\nu] \Downarrow v$	evaluate a closure to a value
$\rho\nu \Downarrow \nu'$	evaluate a substitution closure to an environment
$\omega[\nu] \Downarrow \psi$	evaluate a telescope substitution closure into a telescope environment
$\Omega \Downarrow \Xi$	evaluate a telescope context into a linear telescope context
$\nu(x) \rightsquigarrow v$	look up the value of a variable in an environment
$t^i \rightsquigarrow t'$	one step in calculating the meaning of a lifted term
$\rho^i \rightsquigarrow \rho'$	one step in calculating the meaning of a lifted substitution
$\omega^i \rightsquigarrow \omega'$	one step in calculating the meaning of a lifted telescope substitution
$\nu +\!\!+ \psi \rightsquigarrow \nu'$	composition of environments and telescope environments
$\psi +\!\!+ \psi' \rightsquigarrow \psi''$	composition of telescope environments

We give a case for each judgment type for each constructor. Evaluation of terms is standard, the interesting cases are for t^i and $t^{\textcircled{i}}$: for the former we use the one-step evaluation relation $t^i \rightsquigarrow t'$, for the latter we use the computation rule for $-^{\textcircled{i}}$.

$\boxed{t[\nu] \Downarrow v}$

$$\frac{\rho\nu \Downarrow \nu' \quad t[\nu'] \Downarrow v}{t[\rho][\nu] \Downarrow v} \qquad \frac{\nu(x) \rightsquigarrow v}{x[\nu] \Downarrow v} \qquad \mathsf{U}[\nu] \Downarrow \mathsf{U} \qquad (\Pi(x : X)_I[\omega].A)[\nu] \Downarrow (\Pi(x : X)_I[\omega].A)[\nu]$$

$$(\lambda(x)_I.t)[\nu] \Downarrow (\lambda(x)_I.t)[\nu] \qquad \frac{t[\nu] \Downarrow (\lambda(x)_I.t')[\nu'] \quad \omega[\nu] \Downarrow \psi \quad \nu' +\!\!+ \psi \rightsquigarrow \nu'' \quad t'[\nu''] \Downarrow v}{\mathsf{app}_I(t, \omega)[\nu] \Downarrow v}$$

$$\frac{t[\nu] \Downarrow n \quad \omega[\nu] \Downarrow \psi}{\mathsf{app}_I(t, \omega)[\nu] \Downarrow \mathsf{app}_I(n, \psi)} \qquad \frac{t^i \rightsquigarrow t' \quad t'[\nu] \Downarrow v}{t^i[\nu] \Downarrow v} \qquad \frac{t^i[\mathsf{R}_i][\nu] \Downarrow v}{t^{\textcircled{i}}[\nu] \Downarrow v}$$

For substitutions, the interesting cases are again for our special constructions: the substitutions $\mathsf{0}_i, \mathsf{1}_i$ (denoted b_i) project out the corresponding components, while the substitution R_i duplicates the content of the environment. Its substitution rule ensures that we can move the $-^{\textcircled{i}}$ through the environment ν'.

$\boxed{\rho\nu \Downarrow \nu'}$

$$\mathsf{id}\nu \Downarrow \nu \qquad ()\nu \Downarrow () \qquad \frac{\rho\nu \Downarrow \nu'}{(\rho, x \mapsto t)\nu \Downarrow (\nu', x \mapsto t[\nu])} \qquad \frac{\rho'\nu \Downarrow \nu' \quad \rho\nu' \Downarrow \nu''}{(\rho\rho')\nu \Downarrow \nu''}$$

$$\frac{\rho\nu \Downarrow \nu' \qquad \omega[\nu] \Downarrow \psi \quad \nu' +\!\!+ \psi \rightsquigarrow \nu''}{(\rho +\!\!+ \omega)\nu \Downarrow \nu''}$$

$$b_i() \Downarrow () \qquad \frac{b_i\nu \Downarrow \nu'}{b_i(\nu, x_{ib} \mapsto t) \Downarrow (\nu', x \mapsto t)} \qquad \frac{b_i\nu \Downarrow \nu'}{b_i(\nu, x_{ic} \mapsto t) \Downarrow \nu'}\ {\scriptstyle c \neq b}$$

$$\mathsf{R}_i() \Downarrow () \qquad \frac{\mathsf{R}_i\nu \Downarrow \nu'}{\mathsf{R}_i(\nu, x \mapsto g) \Downarrow (\nu', x_{i0} \mapsto g, x_{i1} \mapsto g, x_{i2} \mapsto g^{\textcircled{i}})}$$

$$\frac{\mathsf{R}_i\nu \Downarrow \nu''}{\mathsf{R}_i(\nu, x \mapsto t[\nu']) \Downarrow (\nu'', x_{i0} \mapsto t[\nu'], x_{i1} \mapsto t[\nu'], x_{i2} \mapsto t^{\textcircled{i}}[\nu'])}$$

$$\frac{\rho^i \rightsquigarrow \rho' \quad \rho'\nu \Downarrow \nu'}{(\rho)^i\nu \Downarrow \nu'} \qquad \frac{\rho^i \rightsquigarrow \rho' \quad \rho'\nu \Downarrow (\nu', x \mapsto t)}{\{\rho\}_i\nu \Downarrow \nu'}$$

Evaluating telescope substitutions follows the same pattern as substitutions. The variable lookup $\nu(x) \rightsquigarrow v$ is standard. The one-step part of the evaluation is given by following the rules in section 3.5:

$\boxed{t^i \rightsquigarrow t'}$

$$
\begin{aligned}
(t[\rho])^i &\rightsquigarrow t^i[\rho^i] \\
x^i &\rightsquigarrow x_{i2} \\
\mathsf{U}^i &\rightsquigarrow \lambda\{X\}_i.\Pi(x : X)_i[\{X \mapsto X\}_i].\mathsf{U} \\
(\Pi(x : X)^I[\omega].A)^i &\rightsquigarrow \lambda\{f\}_i.\Pi(x : X)^{Ii}[\omega^i].\mathsf{app}_i(A^i, \{x' \mapsto \mathsf{app}^I(f, (x'' \mapsto x)^I)\}_i) \\
(\lambda(x)^I.t)^i &\rightsquigarrow \lambda(x)^{Ii}.t^i \\
\mathsf{app}^I(t, \omega)^i &\rightsquigarrow \mathsf{app}^{Ii}(t^i, \omega^i) \\
(\Pi\{x : X\}_I[\omega].\mathsf{U})^i &\rightsquigarrow \lambda(X')_i.\Pi\{x : X\}_{Ii}[(\omega^i, \{X_{I2} \mapsto X'\}_i)].\mathsf{U} \\
(\lambda\{x\}_I.A)^i &\rightsquigarrow \lambda\{x\}_{Ii}.\mathsf{app}_i(A^i, \{x' \mapsto x_{I2}\}_i) \\
\mathsf{app}_I(t, \omega)^i &\rightsquigarrow \lambda\{x'\}_i.\mathsf{app}_{Ii}(t^i, \{\omega^i, (x_{I2} \mapsto x'\}_i))
\end{aligned}
$$

$$
\frac{t^j \rightsquigarrow t'}{(t^j)^i \rightsquigarrow t'^i} \qquad (t\textcircled{j})^i \rightsquigarrow (t^j[\mathsf{R}_j])^i
$$

The definition of $\rho^i \rightsquigarrow \rho'$ for substitutions follows a similar pattern, it uses the functor laws for substitutions given in section 3.4. We also refer to the appendix for the listing of $\omega^i \rightsquigarrow \omega'$, the composition of substitutions and telescopes and for full normalisation which is the recursive application of evaluation.

5 Capturing univalence

We show how to extend the above theory with Σ-types, natural numbers, empty and singleton types. We only list the rules regarding $-^i$ and use pattern matching syntax for clarity:

$$
\begin{aligned}
\mathsf{app}_i((\Sigma(x : A).B)^i, \{w\}_i) &\equiv \Sigma(p : \mathsf{app}_i(A^i, \{\pi_0\, w\}_i)).\mathsf{app}_i(B^i[\{x \mapsto \pi_0\, w\}_i, x_{i2} \mapsto p], \{\pi_1\, w\}_i) \\
(a, b)^i &\equiv (a^i, b^i) \\
(\pi_b\, w)^i &\equiv \pi_b\, w^i \\
\mathsf{app}_i(\mathbb{N}^i, (\mathsf{zero}, \mathsf{zero})) &\equiv 1 \\
\mathsf{app}_i(\mathbb{N}^i, (\mathsf{suc}\, m, \mathsf{zero})) &\equiv 0 \\
\mathsf{app}_i(\mathbb{N}^i, (\mathsf{zero}, \mathsf{suc}\, n)) &\equiv 0 \\
\mathsf{app}_i(\mathbb{N}^i, (\mathsf{suc}\, m, \mathsf{suc}\, n)) &\equiv \mathsf{app}_i(\mathbb{N}^i, (m, n)) \\
\mathsf{app}_i(0^i, \{t\}_i) &\equiv 1 \\
\mathsf{app}_i(1^i, \{t\}_i) &\equiv 1 \\
*^i &\equiv *
\end{aligned}
$$

The idea is that for a type A, the relation A^i can be viewed as its heterogeneous equality relation. However, its definition for the universe is not the right one: two types shouldn't be equal if they are related; they should be equal if they are related by an equivalence relation.

We will use the following, symmetric definition of equivalence[3]:

$$\begin{aligned}(A \simeq B) = \Sigma(R : A \to B \to U)\\ .\Pi(x : A).\mathsf{isContr}(\Sigma(y : B).R\,x\,y) \times \Pi(y : B).\mathsf{isContr}(\Sigma(x : A).R\,x\,y)\end{aligned}$$

This can be expanded to:

$$\begin{aligned}\sim_{\mathsf{U}^i} \equiv \lambda\{X\}_i.\Sigma \sim_{X_{i2}} &: \Pi\{x : X\}_i.\mathsf{U}\\ \overrightarrow{\mathsf{co}}_{X_{i2}} &: \Pi(x_{i0} : X_{i0}).\Sigma(x_{i1} : X_{i1}).x_{i0} \sim_{X_{i2}} x_{i1}\\ \overrightarrow{\mathsf{un}}_{X_{i2}} &: \Pi(x : X)^i.\overrightarrow{\mathsf{co}}_{X_{i2}}\,x_{i0} \sim_{(\Sigma(x_{i1}:X_{i1}).x_{i0}\sim_{X_{i2}}x_{i1})\textcircled{\scriptsize j}} (x_{i1}, x_{i2})\\ \overleftarrow{\mathsf{co}}_{X_{i2}} &: \Pi(x_{i1} : X_{i1}).\Sigma(x_{i0} : X_{i0}).x_{i0} \sim_{X_{i2}} x_{i1}\\ \overleftarrow{\mathsf{un}}_{X_{i2}} &: \Pi(x : X)^i.\overleftarrow{\mathsf{co}}_{X_{i2}}\,x_{i1} \sim_{(\Sigma(x_{i0}:X_{i0}).x_{i0}\sim_{X_{i2}}x_{i1})\textcircled{\scriptsize j}} (x_{i0}, x_{i2})\end{aligned}$$

$a \sim_R b$ is an abbreviation for $\mathsf{app}_i(R, (x_{i0} \mapsto a, x_{i1} \mapsto b))$ where $R : \Pi\{x : X\}_i[\xi].\mathsf{U}$ for some ξ. co (coerce) gives the center of contraction, and un (uncoerce) is the contraction. They need to be defined both directions (by omitting one direction, we would get directed type theory). Note that the un operations uses a new dimension j.

Now we can represent the usual homogeneous equality as $=_A$ as $\sim_{A\textcircled{\scriptsize i}}$ for some fixed but arbitrary i.

As $\mathsf{app}_i(\mathsf{U}^i, (A, B))$ is not just relation space anymore, we need to project out the relation, this is what $\sim$ is doing. Rule 1, which expressed parametricity now becomes the rule which says that terms respect equality:

$$\frac{\Gamma \vdash t : A}{\Gamma^i \vdash t^i : \mathsf{app}_i(\sim_{A^i}, \{x \mapsto t\}_i)}$$

The rules given in section 3.5 mentioning types need to be replaced by ones only defining the $\sim$ component. The other cases remain the same:

$$\begin{aligned}\sim_{\mathsf{U}^i} &\equiv \lambda\{X\}_i.\Pi\{x : X\}_i.\mathsf{U}\\ \sim_{(\Pi(x:X)^I[\xi].B)^i} &\equiv \lambda\{f\}_i.\Pi(x : X)^{Ii}[\xi^i].\mathsf{app}_i(\sim_{B^i}, \{y \mapsto \mathsf{app}^I(f, (x)^I)\}_i)\\ \sim_{(\Pi\{x:X\}_I[\xi].\mathsf{U})^i} &\equiv \lambda\{y\}_i.\Pi\{x : X\}_{Ii}[\xi^i +\!\!+ \{X_{I2} \mapsto y\}_i].\mathsf{U}\\ (\lambda\{x\}_I.B)^i &\equiv \lambda\{x\}_{Ii}.\mathsf{app}_i(\sim_{B^i}, \{y \mapsto x_{I2}\}_i)\\ \sim_{(\mathsf{app}_I(R,\omega))^i} &\equiv \lambda\{y\}_i.\mathsf{app}_{Ij}(R^i, \omega^i +\!\!+ \{x_{I2} \mapsto y\}_i)\end{aligned}$$

We now need to define the operations co and un in both directions for all types – we haven't completed this yet. Once this is done we can derive J from $\mathsf{transport}$ and the proof that singletons are contractible. We derive $\mathsf{transport}$ as follows:

$$\frac{\Gamma \vdash P : A \to \mathsf{U} \quad q : x =_A y \quad u : \mathsf{app}(P, x)}{\Gamma \vdash \mathsf{transport}_P\,q\,u :\equiv \pi_0(\overrightarrow{\mathsf{co}}_{\mathsf{app}^i(P\textcircled{\scriptsize i},(x,y,q))}\,u) : \mathsf{app}(P, y)}$$

We don't know whether this operator will satisfy the β-rule for J definitionally.

[3] Exercise 4.2 in [9] asks to prove that it is equivalent to the usual definition of equivalence. The solution is available in the repository `http://github.com/HoTT/book`.

6 Conclusions and further work

On our quest towards a computational understanding of univalence we have reached a first milestone in being able to present the theory of relational parametricity using a substitution calculus and a big-step normalisation function. We still have to establish that this normalisation function is sound and complete following [1]. We have already a partial definition of co and un and are optimistic that we can complete this. We also investigate truncated versions of the theory which should be easier to capture. This way we can recover a version of Observational Type Theory [2], which does not rely on a definitionally proof-irrelevant universe of propositions. Even once we have captured univalence, the interpretation of higher inductive types poses yet another challenge.

The use of dimensions with names is reminiscent of [4], however in this calculus all judgments are decorated with a set of dimensions and they don't use a substitution calculus and hence $\mathsf{0}, \mathsf{1}, \mathsf{R}$ are operations on the syntax and not substitutions as in our presentation.

References

1 Thorsten Altenkirch and James Chapman. Big-step normalisation. *Journal of Functional Programming*, 19(3-4):311–333, 2009.

2 Thorsten Altenkirch, Conor Mcbride, and Wouter Swierstra. Observational equality, now! In *PLPV '07: Proceedings of the 2007 workshop on Programming languages meets program verification*, pages 57–58. ACM, 2007.

3 Robert Atkey, Neil Ghani, and Patricia Johann. A relationally parametric model of dependent type theory. In *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2014)*, 2014.

4 Jean-Philippe Bernardy and Moulin Guilhem. Type-theory in color. In *Proceedings of the 18th ACM SIGPLAN International Conference on Functional Programming*, ICFP '13, pages 61–72, New York, NY, USA, 2013. ACM.

5 Jean-Philippe Bernardy and Guilhem Moulin. A computational interpretation of parametricity. In *Proceedings of the 2012 27th Annual IEEE/ACM Symposium on Logic in Computer Science*, LICS '12, pages 135–144, Washington, DC, USA, 2012. IEEE Computer Society.

6 Marc Bezem, Thierry Coquand, and Simon Huber. A Model of Type Theory in Cubical Sets. In Ralph Matthes and Aleksy Schubert, editors, *19th International Conference on Types for Proofs and Programs (TYPES 2013)*, volume 26 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 107–128, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

7 James Chapman. Type theory should eat itself. *Electron. Notes Theor. Comput. Sci.*, 228:21–36, January 2009.

8 Claudio Hermida, Uday S. Reddy, and Edmund P. Robinson. Logical relations and parametricity – a Reynolds programme for category theory and programming languages. *Electronic Notes in Theoretical Computer Science*, 303(0):149 – 180, 2014. Proceedings of the Workshop on Algebra, Coalgebra and Topology (WACT 2013).

9 The Univalent Foundations Program. Homotopy type theory: Univalent foundations of mathematics. Technical report, Institute for Advanced Study, 2013.

10 J. C. Reynolds. Types, abstraction and parametric polymorphism. In R. E. A. Mason, editor, *Information Processing 83, Proceedings of the IFIP 9th World Computer Congress, Paris, September 19-23, 1983*, pages 513–523. Elsevier Science Publishers B. V. (North-Holland), Amsterdam, 1983.

A All rules of the operational semantics

A.1 Evaluation

t, A	$::= t[\rho] \mid x \mid \mathsf{U} \mid \Pi(x : X)^I[\omega].A \mid \Pi\{x : X\}_I[\omega].\mathsf{U}$	
	$\mid \lambda\{\!\{x\}\!\}_I.t \mid \mathsf{app}_I(t, \omega) \mid t^i \mid t^{\textcircled{i}}$	terms
ρ	$::= \mathsf{id} \mid () \mid (\rho, x \mapsto t) \mid \rho\rho' \mid \rho +\!\!+ \omega \mid \mathsf{0}_i \mid \mathsf{1}_i \mid \mathsf{R}_i \mid \rho^i \mid \{\rho\}_i$	substitutions
ω	$::= () \mid (\omega, x \mapsto t) \mid \omega[\rho] \mid \omega +\!\!+ \omega' \mid \mathsf{pr}_\Omega \mid \omega^i \mid \{\omega\}_i$	telescope substitutions
x, f, X	$::= \ldots$	variables
i	$::= \ldots$	dimension names
I	$::= \emptyset \mid Ii$	sets of dimension names
v	$::= \mathsf{U} \mid (\Pi\{\!\{x : X\}\!\}_I[\omega].A)[\nu] \mid (\lambda\{\!\{x\}\!\}_I.t)[\nu] \mid n$	values (whnfs)
ν	$::= () \mid (\nu, x \mapsto g) \mid (\nu, x \mapsto t[\nu])$	environments
n	$::= g \mid \mathsf{app}_I(n, \psi)$	neutral values
g	$::= x \mid g^{\textcircled{i}}$	generic values
ψ	$::= () \mid (\psi, x \mapsto g) \mid (\psi, x \mapsto t[\nu])$	telescope environments
Ω	$::= \cdot \mid \Omega.x : A \mid \Omega[\rho] \mid \Omega +\!\!+ \Omega' \mid \Omega^i \mid \{\Omega\}_i$	telescope contexts
Ξ	$::= \cdot \mid \Xi.x : A$	linear telescope contexts

Judgment types:

$t[\nu] \Downarrow v$	evaluate a closure to a value
$\rho\nu \Downarrow \nu'$	evaluate a substitution closure to an environment
$\omega[\nu] \Downarrow \psi$	evaluate a telescope substitution closure into a telescope environment
$\Omega \Downarrow \Xi$	evaluate a telescope context into a linear telescope context
$\nu(x) \rightsquigarrow v$	look up the value of a variable in an environment
$t^i \rightsquigarrow t'$	one step in calculating the meaning of a lifted term
$\rho^i \rightsquigarrow \rho'$	one step in calculating the meaning of a lifted substitution
$\omega^i \rightsquigarrow \omega'$	one step in calculating the meaning of a lifted telescope substitution
$\nu +\!\!+ \psi \rightsquigarrow \nu'$	composition of environments and telescope environments
$\psi +\!\!+ \psi' \rightsquigarrow \psi''$	composition of telescope environments
$\mathsf{pr}_\Xi[\nu] \rightsquigarrow \psi$	evaluating a telescope projection

$\boxed{t[\nu] \Downarrow v}$

$$\frac{\rho\nu \Downarrow \nu' \quad t[\nu'] \Downarrow v}{t[\rho][\nu] \Downarrow v} \qquad \frac{\nu(x) \rightsquigarrow v}{x[\nu] \Downarrow v} \qquad \mathsf{U}[\nu] \Downarrow \mathsf{U} \qquad (\Pi\{\!\{x : X\}\!\}_I[\omega].A)[\nu] \Downarrow (\Pi\{\!\{x : X\}\!\}_I[\omega].A)[\nu]$$

$$(\lambda\{\!\{x\}\!\}_I.t)[\nu] \Downarrow (\lambda\{\!\{x\}\!\}_I.t)[\nu] \qquad \frac{t[\nu] \Downarrow (\lambda(x)_I.t')[\nu'] \quad \omega[\nu] \Downarrow \psi \quad \nu' +\!\!+ \psi \rightsquigarrow \nu'' \quad t'[\nu''] \Downarrow v}{\mathsf{app}_I(t, \omega)[\nu] \Downarrow v}$$

$$\frac{t[\nu] \Downarrow n \quad \omega[\nu] \Downarrow \psi}{\mathsf{app}_I(t, \omega)[\nu] \Downarrow \mathsf{app}_I(n, \psi)} \qquad \frac{t^i \rightsquigarrow t' \quad t'[\nu] \Downarrow v}{t^i[\nu] \Downarrow v} \qquad \frac{t^i[\mathsf{R}_i][\nu] \Downarrow v}{t^{\textcircled{i}}[\nu] \Downarrow v}$$

$\boxed{\rho\nu \Downarrow \nu'}$

$$\mathsf{id}\nu \Downarrow \nu \qquad ()\nu \Downarrow () \qquad \frac{\rho\nu \Downarrow \nu'}{(\rho, x \mapsto t)\nu \Downarrow (\nu', x \mapsto t[\nu])} \qquad \frac{\rho'\nu \Downarrow \nu' \quad \rho\nu' \Downarrow \nu''}{(\rho\rho')\nu \Downarrow \nu''}$$

$$\frac{\rho\nu \Downarrow \nu' \qquad \omega[\nu] \Downarrow \psi \quad \nu' +\!\!+ \psi \rightsquigarrow \nu''}{(\rho +\!\!+ \omega)\nu \Downarrow \nu''}$$

$$b_i() \Downarrow () \qquad \frac{b_i\nu \Downarrow \nu'}{b_i(\nu, x_{ib} \mapsto t) \Downarrow (\nu', x \mapsto t)} \qquad \frac{b_i\nu \Downarrow \nu'}{b_i(\nu, x_{ic} \mapsto t) \Downarrow \nu'}\ {\scriptstyle c \neq b}$$

$$\mathsf{R}_i() \Downarrow () \qquad \frac{\mathsf{R}_i\nu \Downarrow \nu'}{\mathsf{R}_i(\nu, x \mapsto g) \Downarrow (\nu', x_{i0} \mapsto g, x_{i1} \mapsto g, x_{i2} \mapsto g^{\textcircled{i}})}$$

$$\frac{\mathsf{R}_i\nu \Downarrow \nu''}{\mathsf{R}_i(\nu, x \mapsto t[\nu']) \Downarrow (\nu'', x_{i0} \mapsto t[\nu'], x_{i1} \mapsto t[\nu'], x_{i2} \mapsto t^{\textcircled{i}}[\nu'])}$$

$$\frac{\rho^i \rightsquigarrow \rho' \quad \rho'\nu \Downarrow \nu'}{(\rho)^i\nu \Downarrow \nu'} \qquad \frac{\rho^i \rightsquigarrow \rho' \quad \rho'\nu \Downarrow (\nu', x \mapsto t)}{\{\rho\}_i\nu \Downarrow \nu'}$$

$\boxed{\omega[\nu] \Downarrow \psi}$

$$()[\nu] \Downarrow () \qquad \frac{\omega[\nu] \Downarrow \psi}{(\omega, x \mapsto t)[\nu] \Downarrow (\psi, x \mapsto t[\nu])} \qquad \frac{\rho\nu \Downarrow \nu' \quad \omega[\nu'] \Downarrow \psi}{\omega[\rho][\nu] \Downarrow \psi}$$

$$\frac{\omega[\nu] \Downarrow \psi \quad \omega'[\nu] \Downarrow \psi \quad \psi +\!\!+ \psi' \rightsquigarrow \psi''}{(\omega +\!\!+ \omega')[\nu] \Downarrow \psi''} \qquad \frac{\Omega \Downarrow \Xi \quad \mathsf{pr}_\Xi \rightsquigarrow \psi}{\mathsf{pr}_\Omega[\nu] \Downarrow \psi} \qquad \frac{\omega^i \rightsquigarrow \omega' \quad \omega'[\nu] \Downarrow \psi}{\omega^i[\nu] \Downarrow \psi}$$

$$\frac{\omega^i \rightsquigarrow \omega' \quad \omega'[\nu] \Downarrow (\psi, x \mapsto t)}{\{\omega\}_i[\nu] \Downarrow \psi}$$

$\boxed{\Omega \Downarrow \Xi}$

$$\cdot \Downarrow \cdot \qquad \frac{\Omega \Downarrow \Xi}{\Omega.x : A \Downarrow \Xi.x : A} \qquad \frac{\Omega \Downarrow \cdot}{\Omega[\rho] \Downarrow \cdot} \qquad \frac{\Omega \Downarrow \Xi.x : A \quad \Xi[\rho] \Downarrow \Xi'}{\Omega[\rho] \Downarrow \Xi'.x : A[\rho]}$$

$$\frac{\Omega \Downarrow \cdot \qquad \Omega' \Downarrow \Xi}{\Omega +\!\!+ \Omega' \Downarrow \Xi} \qquad \frac{\Omega' \Downarrow (\Xi.x : A) \qquad \Omega +\!\!+ \Xi \Downarrow \Xi'}{\Omega +\!\!+ \Omega' \Downarrow \Xi'.x : A} \qquad \frac{\Omega \Downarrow \cdot}{\Omega^i \Downarrow \cdot}$$

$$\frac{\Omega \Downarrow \Xi.x : A \qquad \Xi^i \Downarrow \Xi'}{\Omega^i \Downarrow \Xi'.x_{i0} : A[\mathsf{0}_i].x_{i1} : A[\mathsf{1}_i].x_{i2} : \mathsf{app}_i(A^i, (x \mapsto x)_i)} \qquad \frac{\Omega \Downarrow \Xi.x : A \qquad \Xi^i \Downarrow \Xi'}{\{\Omega\}_i \Downarrow \Xi'.x_{i0} : A[\mathsf{0}_i].x_{i1} : A[\mathsf{1}_i]}$$

$\boxed{\nu(x) \rightsquigarrow v}$

$$\frac{\nu(x) = g}{\nu(x) \rightsquigarrow g} \qquad \frac{\nu(x) = t[\nu'] \quad t[\nu'] \Downarrow v}{\nu(x) \rightsquigarrow v}$$

$\boxed{t^i \rightsquigarrow t'}$

$$
\begin{aligned}
(t[\rho])^i &\rightsquigarrow t^i[\rho^i] \\
x^i &\rightsquigarrow x_{i2} \\
\mathsf{U}^i &\rightsquigarrow \lambda\{X\}_i.\Pi(x:X)_i[\{X \mapsto X\}_i].\mathsf{U} \\
(\Pi(x:X)^I[\omega].A)^i &\rightsquigarrow \lambda\{f\}_i.\Pi(x:X)^{Ii}[\omega^i].\mathsf{app}_i(A^i, \{x' \mapsto \mathsf{app}^I(f, (x'' \mapsto x)^I)\}_i) \\
(\lambda(x)^I.t)^i &\rightsquigarrow \lambda(x)^{Ii}.t^i \\
\mathsf{app}^I(t,\omega)^i &\rightsquigarrow \mathsf{app}^{Ii}(t^i, \omega^i) \\
(\Pi\{x:X\}_I[\omega].\mathsf{U})^i &\rightsquigarrow \lambda(X')_i.\Pi\{x:X\}_{Ii}[(\omega^i, \{X_{I2} \mapsto X'\}_i)].\mathsf{U} \\
(\lambda\{x\}_I.A)^i &\rightsquigarrow \lambda\{x\}_{Ii}.\mathsf{app}_i(A^i, \{x' \mapsto x_{I2}\}_i) \\
\mathsf{app}_I(t,\omega)^i &\rightsquigarrow \lambda\{x'\}_i.\mathsf{app}_{Ii}(t^i, \{\omega^i, (x_{I2} \mapsto x'\}_i))
\end{aligned}
$$

$$\frac{t^j \rightsquigarrow t'}{(t^j)^i \rightsquigarrow t'^i} \qquad (t\textcircled{j})^i \rightsquigarrow (t^j[\mathsf{R}_j])^i$$

$\boxed{\rho^i \rightsquigarrow \rho'}$

$$
\begin{aligned}
\mathsf{id}^i &\rightsquigarrow \mathsf{id} \\
()^i &\rightsquigarrow () \\
(\rho, x \mapsto t)^i &\rightsquigarrow (\rho^i, x_{i0} \mapsto t[\mathsf{0}_i], x_{i1} \mapsto t[\mathsf{1}_i], x_{i2} \mapsto t^i) \\
(\rho\rho')^i &\rightsquigarrow \rho^i \rho'^i \\
(\rho +\!\!+ \omega)^i &\rightsquigarrow (\rho^i +\!\!+ \omega^i) \\
(b_j)^i &\rightsquigarrow b_j \\
(\mathsf{R}_j)^i &\rightsquigarrow \mathsf{R}_j
\end{aligned}
$$

$$\frac{\rho^j \rightsquigarrow \rho'}{((\rho)_j)^i \rightsquigarrow \rho'^i}$$

$\boxed{\omega^i \rightsquigarrow \omega'}$

$$
\begin{aligned}
()^i &\rightsquigarrow () \\
(\omega, x \mapsto t)^i &\rightsquigarrow (\omega^i, x_{i0} \mapsto t[\mathsf{0}_i], x_{i1} \mapsto t[\mathsf{1}_i], x_{i2} \mapsto t^i) \\
(\omega[\rho])^i &\rightsquigarrow \omega^i[\rho^i] \\
(\omega +\!\!+ \omega')^i &\rightsquigarrow (\omega^i +\!\!+ \omega'^i)
\end{aligned}
$$

$$\frac{\omega^j \rightsquigarrow \omega'}{((\omega)_j)^i \rightsquigarrow \omega'^i}$$

$\boxed{\nu +\!\!+ \psi \rightsquigarrow \nu'}$

$$\nu +\!\!+ () \rightsquigarrow \nu \qquad \frac{\nu +\!\!+ \psi \rightsquigarrow \nu'}{\nu +\!\!+ (\psi, x \mapsto t) \rightsquigarrow (\nu', x \mapsto t)}$$

$\boxed{\psi +\!\!+ \psi' \rightsquigarrow \psi''}$

$$\psi +\!\!+ () \rightsquigarrow \psi \qquad \frac{\psi +\!\!+ \psi' \rightsquigarrow \psi''}{\psi +\!\!+ (\psi', x \mapsto t) \rightsquigarrow (\psi'', x \mapsto t)}$$

$\boxed{\mathsf{pr}_\Xi[\nu] \rightsquigarrow \psi}$

$$\mathsf{pr}_\cdot[\nu] \Downarrow () \qquad \frac{\mathsf{pr}_\Xi[\nu] \Downarrow \psi}{\mathsf{pr}_{\Xi.x:A}[(\nu, x \mapsto t)] \Downarrow (\psi, x \mapsto t)}$$

A.2 Normalisation

$$\begin{array}{lll} v_n & ::= \mathsf{U} \mid \Pi\{\!\{x : X\}\!\}_I[\psi_n].v_n \mid \lambda\{\!\{x\}\!\}_I.v_n \mid n_n & \text{normal forms} \\ n_n & ::= g \mid \mathsf{app}_I(n, \psi_n) & \text{neutral normal forms} \\ \psi_n & ::= () \mid (\psi_n, x \mapsto v_n) & \text{normal telescopes} \end{array}$$

$\boxed{v \Rrightarrow v_n}$

$$\mathsf{U} \Rrightarrow \mathsf{U}$$

$$\frac{\{\!\{x : X\}\!\}_I[\mathsf{id} +\!\!+ \omega] \Downarrow \Xi \quad \mathsf{pr}_\Xi \rightsquigarrow \psi \quad \nu +\!\!+ \psi \rightsquigarrow \nu' \quad A[\nu'] \Downarrow v \quad v \Rrightarrow v_n \quad \omega[\nu] \Downarrow \psi' \quad \psi' \Rrightarrow \psi'_n}{(\Pi\{\!\{x : X\}\!\}_I[\omega].A)[\nu] \Rrightarrow \Pi\{\!\{x : X\}\!\}_I[\psi'_n].v_n}$$

$$\frac{\{\!\{x : X\}\!\}_I[\mathsf{id} +\!\!+ \omega] \Downarrow \Xi \quad \mathsf{pr}_\Xi \rightsquigarrow \psi \quad \nu +\!\!+ \psi \rightsquigarrow \nu' \quad t[\nu'] \Downarrow v \quad v \Rrightarrow v_n}{(\lambda\{\!\{x\}\!\}_I.t)[\nu] \Rrightarrow \lambda\{\!\{x\}\!\}_I.v_n} \qquad \frac{n \Rrightarrow n_n}{n \Rrightarrow n_n}$$

$\boxed{n \Rrightarrow n_n}$

$$g \Rrightarrow g \qquad \frac{n \Rrightarrow n_n \quad \psi \Rrightarrow \psi_n}{\mathsf{app}_I(n, \psi) \Rrightarrow \mathsf{app}_I(n_n, \psi_n)}$$

$\boxed{\psi \Rrightarrow \psi_n}$

$$() \Rrightarrow () \qquad \frac{\psi \Rrightarrow \psi_n}{(\psi, x \mapsto g) \Rrightarrow (\psi_n, x \mapsto g)} \qquad \frac{\psi \Rrightarrow \psi_n \quad t[\nu] \Downarrow v \quad v \Rrightarrow v_n}{(\psi, x \mapsto t[\nu]) \Rrightarrow (\psi_n, x \mapsto v_n)}$$

To generate the identity substitution, we need to linearize contexts, it can be done in the same way as for telescope contexts before ($\Omega \Downarrow \Xi$). We denote this by $\Gamma \Downarrow \Delta$.

$\boxed{\mathsf{id}_\Delta \Downarrow \nu}$

$$\mathsf{id}_\cdot \Downarrow () \qquad \frac{\mathsf{id}_\Delta \Downarrow \nu}{\mathsf{id}_{\Delta.x:t} \Downarrow (\nu, x \mapsto x)}$$

Normalisation can be performed as follows:

$$\frac{\Gamma \vdash t : A \qquad \Gamma \Downarrow \Delta \qquad \mathsf{id}_\Delta \Downarrow \nu \qquad t[\nu] \Downarrow v \qquad v \Rrightarrow v_n}{t \text{ normalises to } v_n}$$