Equideductive Categories and their Logic

Paul Taylor

22 August 2013

1 Introduction

In this paper we study the interaction between equalisers and exponentials. Since these are both defined as *right* adjoints, their universal properties can be combined into a single one, so it is surprising that something as elementary as this seems not to have been investigated before in the categorical literature. Of course these structures co-exist in a topos or a locally cartesian closed category, but we are going to study less powerful structures with a view to managing the complexity that arises from this interaction. The equideductive categories that we shall introduce are not themselves cartesian closed but are embedded "in a nice way" in cartesian closed extensions.

Notation 1.1 We would like to write

$$\{x : X \mid \forall y{:}Y.\ \alpha xy = \beta xy\} \rightarrowtail X \underset{\beta}{\overset{\alpha}{\rightrightarrows}} \Sigma^Y$$

for an equaliser targeted at an exponential. Then, if Y is itself the equaliser of $\gamma, \delta : B \rightrightarrows \Sigma^Z$, we also want to use the notation recursively, for example

$$\{x : X \mid \forall y{:}B.\ (\forall z{:}Z.\ \gamma yz = \delta yz) \Longrightarrow \alpha xy = \beta xy\}.$$

The principal objective of this paper is to introduce a symbolic calculus that justifies these expressions. It is set out in Section 9.

This is an *external* logic of subobjects (constructions with equalisers *etc.*) that can be formulated in a suitable category. It does not assume that the category has a subobject *classifier* with its internal logic. Whilst the key object Σ will become the Sierpiński space or the subobject classifier in the leading examples of topology, set theory or recursion theory, the agreement between the internal and external logics must obviously depend on some additional property of Σ. We shall study this in a later paper [DD].

The universal property of the equaliser above can be expressed in a way that involves the product $X \times Y$ instead of the exponential Σ^Y. In the absence of a better name, we call this a "partial product". We show in the next section how it captures infinitary intersections and universal quantification, as well as equalisers of exponentials in cartesian closed super-categories such as the Yoneda embedding.

Notice, before we begin to formalise Notation 1.1 as part of a new calculus, that these expressions with $\forall$ and $\Rightarrow$ are less general than we are accustomed to using elsewhere in logic:

Definition 1.2 Any expression $\forall y.\ \mathsf{q}(y) \Rightarrow \alpha xy = \beta xy$ that arises from equalisers of exponentials obeys the ***variable-binding rule***:

every free variable that occurs in the ***antecedents*** (*i.e.* on the *left*) of $\Rightarrow$

must be bound by the quantifier.

Variables that only occur on the *right* of $\Rightarrow$ may remain free.

The reason why this happens is that the variables range over objects X and Y that are themselves *fixed* types, instead of being type-*expressions* that involve variables that range over other types. A type theorist or categorical logician may object that we are just being lazy here. The former would rewrite the fixed type Y as a dependent one $Y[x]$, whilst the latter would implement this by replacing the product projection $\pi_0 : X \times Y \to X$ with a arbitrary map in the definition of a partial product.

We freely confess that we would like to avoid dependent types until we fully understand the situation without them. However, there are also more substantial reasons for adopting this rule, namely that such changes would yield a theory that is only suitable for set theory and not topology or recursion. The relevant counterexample is given in [BB], which also shows that we cannot ask for exponentials in every slice category, or dependent products with arbitrary parameters.

In fact, some of the results in our theory are *only* valid for predicates that satisfy the variable-binding rule. This is the case in particular for Lemma 10.12 here and for the properties of the "existential quantifier" $\ni$ that we introduce in [BB]. On the other hand, it will be no handicap in [?], where we show that any equideductive category may be embedded in a very simple way into a cartesian closed one, because that construction will only use expressions that obey this rule.

Somewhat miraculously, several of the limitations that the rule imposes on the expressivity of the logic will be relaxed in [DD], where we add the lattice structure to Σ and require its order relation to agree with equideductive implication ($\Rightarrow$). For example, we find that there that variables whose type is discrete or Hausdorff are exempt from the variable-binding rule. This means that we shall not even need to modify this rule in our basic definition of partial products when we characterise set theory (*i.e.* the situation in which the category is an elementary topos) using equideductive logic.

Remark 1.3 Even though we do not ask that Σ be a lattice in this paper, we shall include some of the other categorical properties of the Sierpiński space and the subobject classifier in the definition of an equideductive category in Section 4, besides the interaction of equalisers of exponentials.

Consider the nested use of Notation 1.1 above. Although the main *equation* $\alpha xy = \beta xy$ is only intended to place a restriction on x for each of those y for which $\forall z.\, \gamma yz = \delta yz$, the *terms* αxy and βxy must still be *well formed* for all y of type B.

More generally, when we write a program αxy, we intend it to behave in a certain way so long as y belongs to some subspace Y of legal input values. Nevertheless, this program will do *something* given *any* input value of y of its syntactic type B, even if this is to print an error message or to do something disastrous.

This property has long been familiar in general topology and elsewhere:

Definition 1.4 An object I is ***injective*** with respect to the map $i : U \to V$ if, for any map $f : U \to I$, there is some "lifting" $g : V \to I$ with $f = g \cdot i$,

$$\begin{tikzcd} U \arrow[rr, rightarrowtail, "i"] \arrow[dr, "f"'] & & V \arrow[dl, dotted, "g"] \\ & I & \end{tikzcd}$$

In order to justify the nested use of Notation 1.1, we therefore require Σ to be injective with respect to the inclusion $X \times Y \to X \times B$.

Warning 1.5 When the equideductive category $\mathcal{Q}$ is embedded in a cartesian closed category $\mathcal{S}$ with equalisers in [?], we shall just ask for this property for maps in $\mathcal{Q}$, because it will not generalise to $\mathcal{S}$. In particular, we shall not expect Σ to be injective with respect to regular monos in the Yoneda embedding of $\mathcal{Q}$ (Proposition 2.12).

Remark 1.6 The injectives carry all of the computation that we need, so long as there are ***enough*** of them. In the traditional semantic structures, having enough injectives just means that any object can be expressed as an equaliser of them. However, if we look for examples based on recursion theory instead of set theory, the subspaces are not just defined using equations but by predicates with increasingly many alternating quantifiers (Corollary 3.19). This example requires *iterated* expression of objects in terms of injectives that must be reflected in both our categorical definition and the calculus that we introduce here.

Remark 1.7 Something else that we rather took for granted in writing αxy in Notation 1.1 was the λ-calculus. At least, we shall ask for exponentials of the form Σ^A for some sufficiently rich collection of objects A, although not the whole of the category. In topology, we are thinking of A as a locally compact space, whose topology is the exponential Σ^A, which is injective and is a continuous distributive lattice with the Scott topology. In recursion theory there is a similar object to the Sierpiński space that tests termination of programs. Whilst the category of affine varieties over a field has partial products, it does not have these exponentials.

The underlying form of computation using powers of Σ (as opposed to general exponentials) is called the *restricted λ-calculus*. The Abstract Stone Duality programme added lattice structure to this to provide a language for computably based locally compact topological spaces. From a symbolic perspective, injectivity will allow us to continue to use this language for the terms of our new logic. Then equations between such terms will provide the "atomic" predicates, and partial products a logic with $\forall$ and $\Rightarrow$ based on these. Therefore, even though the new programme will take us beyond locally compact spaces, they will still play a central role.

Hence the equideductive world consists of three nested categories, $\mathcal{A} \subset \mathcal{Q} \subset \mathcal{S}$, whose objects loosely play the roles of locally compact, sober and equilogical spaces respectively in the more concrete models that are found in the existing literature. Categorically, $\mathcal{A}$ has exponentials $\Sigma^{(-)}$ but not equalisers, whilst $\mathcal{Q}$ has equalisers but not exponentials, and $\mathcal{S}$ has both.

The category of affine varieties over a field K (the opposite of the category of commutative K-algebras and homomorphisms) has partial products when we take Σ to be the field as a variety (corresponding to the algebra of polynomials $K[x]$ in one variable). There are also enough injectives as below. However, there are not enough exponentials, so maybe some other reification of functions is needed in this application.

Remark 1.8 Having collected these semantic ideas, we can begin to introduce some syntax for them. Section 5 presents the rules of the restricted λ-calculus that provides the object language of equideductive logic. Section 6 shows that these are equivalent to a category with products and exponentials of the form Σ^A. Whilst this material is familiar, we choose slightly different rules from the usual ones, because they will be more convenient in equideductive logic. The equivalence also provides a plan for the same task in the logic. We give the name *urterm* to the λ-terms because we use them as codes, whilst the morphisms of the categories that we construct are partial equivalence classes of urterms. Similarly, the objects of $\mathcal{A}$ are called *urtypes* in the syntax and *urspaces* in a concrete setting.

Section 7 shows that any urspace (exponentiable object) of an equideductive category is *sober* in the abstract sense that was introduced in [A]. That paper also showed that sobriety for $\mathbb{N}$ is equivalent to definition by description, whilst it is Dedekind completeness for $\mathbb{R}$ [I], so this concept is very important for expressing ordinary mathematics. Section 8 builds it in to the λ-calculus using the focus operation.

Remark 1.9 However, the focus operation leads to a lot of complications in the proof theory and other aspects of foundations, so we devote a lot of effort into *removing* them. There are two possible approaches to presenting a new calculus when one of the major goals is then to *eliminate* some of its features. One is to give a *rich* formulation that would be suitable for applications but prove theorems such as *cut elimination* to show that certain parts are redundant. This would, however, create unnecessary difficulties for our main task, which is to interpret the logic in a suitable category and then to show that the two are equivalent.

We shall therefore do the opposite, giving a relatively *poorer* formulation first and then showing how extra features such as focus can be added as *definitional extensions*. Both here and in later work, we take advantage of the minimal syntax to bootstrap mathematical applications in this way.

Those who are experienced in proof theory will be aware that these two ways of proceeding are equivalent. However, our situation is a good deal simpler than the classic case of eliminating cut from the sequent calculus (Gerhard Gentzen's so-called *Hauptsatz*). Each of our definitional extensions (or, if you prefer, elimination lemmas) relates a richer calculus to one that is *obviously* simpler, whilst the lower-level versions of the proofs are not much longer than the ones in the more expressive language. This could, therefore, be seen as a simple one-pass translation that, when implemented on a machine, might be done "on the fly" during analysis of the input syntax.

Specifically, Section 8 only introduces focus axiomatically for *base* types and then defines it for products and exponentials. We also show how to substitute for and into focus-terms and how to apply other terms to them. This culminates in a partial normalisation theorem saying that focus is only needed once, on the outside of a term. Hence the bulk of a computation is expressed using "logical" urterms in the restricted λ-calculus. The focus operation, which is equivalent to definition by description for integers, *drives* this computation by demanding a result that satisfies the relevant property.

Remark 1.10 Section 9 graduates from the λ-calculus object language to equideductive logic itself. It is a kind of predicate calculus with $\forall$, $\Rightarrow$ and &. An extensionality law is used instead of the η- and equality-transmitting rules for the λ-calculus. At first we leave focus out of the object language because it involves more proof-theoretic complications, but we add it back in again and deal with these in Section 10.

Section 11 defines the interpretation of equideductive logic in an equideductive category. Conversely, Section 12 constructs the classifying category from the logic, using injective urtypes for simplicity. Then Section 13 shows how to extend pure equideductive logic to match any given equideductive category.

In Section 14 we introduce a "comprehension" syntax with a curly-bracket notation that makes a type from an urtype with a predicate. This way of introducing subobjects is what ordinary mathematicians seem to mean when they claim that they use "set theory" as the foundation for their subject. We will be able to use the same notation and methods of reasoning, although our predicate calculus is of course much weaker than the usual one.

This notation has numerous mathematical applications. In this paper we use it to complete the discussion of the classifying category and to construct those exponentials that exist. Equideductive logic without focus is a rather feeble fragment of type theory, but with it we have quite a natural proto-set theory, in which sobriety handles functions that are only conditionally defined.

Definition 1.11 Equideductive logic will have its own story to tell about ***equality*** in terms of deductions of equations. Before it does so, we shall need to consider various other notions. We shall therefore say that morphisms of a category are ***the same***, whilst λ-terms are ***interchangeable***.

2 Partial products

The universal property that provides the categorical basis for Notation 1.1 brings together various ideas of intersections of families of subobjects, universally quantified implication and equalisers of exponentials. Throughout, we work in a category $\mathcal{Q}$ that has all finite limits.

Definition 2.1 The map $i : E \to A$ is called (the inclusion of) the ***partial product*** of the parallel pair $\alpha, \beta : A \times Y \rightrightarrows \Sigma$ if
(a) the composites $E \times Y \to A \times Y \rightrightarrows \Sigma$ are the same; and
(b) whenever $a : \Gamma \to A$ is another map for which the composites $\Gamma \times Y \to A \times Y \rightrightarrows \Sigma$ are the same, there is a *unique* map $e : \Gamma \to E$ such that a is the same as the composite $i \cdot e$.

$$\begin{tikzcd}
& E \arrow[dr, "i"] & & \\
\Gamma \arrow[ur, dotted, "e"] \arrow[rr, "a"] & & A & \\
& E \times Y \arrow[uu] \arrow[dr, "i \times Y"] \arrow[drr, bend left] & & \\
\Gamma \times Y \arrow[uu, "\pi_0"] \arrow[ur, dotted, "e \times Y"] \arrow[rr, "a \times Y"] & & A \times Y \arrow[uu, "\pi_0"'] \arrow[r, shift left, "\alpha"] \arrow[r, shift right, "\beta"'] & \Sigma
\end{tikzcd}$$

Although partial products seem rather hybrid at first sight, they have arisen in numerous different contexts. The original one was a notion of dimension in topology, where it explained how the sphere $S^{n+m} \subset \mathbb{R}^{n+m+1}$ is the "product" of spheres S^n and S^m [Pas65]. In categorical type theory, general dependent products (as in a locally cartesian closed category) can be reduced to partial products [Tay99, Thm 9.4.14]. For further background information on this notion, see [Nie82, DT87]

Beware, however, that the property above is a slightly modified special case of the standard formulation. Although we shall use this term throughout this paper, if you wish to refer to the idea, please repeat this warning that our usage is not standard.

We could perhaps call our notion a *partial equaliser* instead, but that would only evade the fact that these ideas really need a better name.

Example 2.2 The notion of partial product can be used to express the "external" intersection of a family of subobjects in a purely categorical way, without using the internal logic of Ω in a topos.

Let $\{U_y \subset A \mid y \in Y\}$ be a family of subobjects of an object A, indexed by another object Y. Getting away from the set-theoretic notion of collection, we can encode this family as a single subobject $V \equiv \{(y, a) \mid a \in U_y\} \subset A \times Y$. Suppose that the inclusion $V \subset A \times Y$ is a regular mono, *i.e.* the equaliser of some pair $\alpha, \beta : A \times Y \rightrightarrows \Sigma$.

Now consider when another subobject $\Gamma \subset A$ is contained in the intersection of the family:

$$\begin{array}{rcl}
\Gamma \subset \bigcap_{y \in Y} U_y & \iff & \forall \gamma \in \Gamma.\, \forall y \in Y.\, \gamma \in U_y \\
& \iff & \forall \gamma \in \Gamma.\, \forall y \in Y.\, (\gamma, y) \in V \\
& \iff & \Gamma \times Y \subset V \\
& \iff & \forall \gamma \in \Gamma.\, \forall y \in Y.\, \alpha(\gamma, y) = \beta(\gamma, y) \\
& \iff & \text{the composites } \Gamma \times Y \to A \times Y \rightrightarrows \Sigma \text{ are the same.}
\end{array}$$

The last line is the hypothesis of clause (b) in the definition of a partial product, so this gives a map $\Gamma \to E$. The intersection $\bigcap U_y$ is the greatest such Γ, namely E, because Γ has the above properties iff $\Gamma \subset E$. This is the special case of the universal property with an inclusion $\Gamma \subset A$ rather than a general map $\Gamma \to A$. $\square$

Motivated by this example, we make the

Definition 2.3 By a ***subspace*** we shall mean a map $i : E \rightarrowtail A$ that (is isomorphic to one that) arises as $i : E \to A$ in some partial product diagram.

Notwithstanding Lemmas 2.14 and 4.7, beware that this map need not in general be expressible as an equaliser $E \hookrightarrow A \rightrightarrows K$. This is because there need not be any suitable target object K (such as the cokernel, *i.e.* the pushout of $E \to A$ against itself) in the category $\mathcal{Q}$. The difference between equalisers and partial products is on of logical complexity, as we shall see from the recursive model in the next section.

We write $\mathcal{M} \subset \mathcal{Q}$ for the class of subspace maps. When we use the word "subspace", we expect this class to be a ***dominion*** [Ros86]:

(a) all isomorphisms should be $\mathcal{M}$-maps;

(b) the maps in $\mathcal{M}$ should be monos;

(c) the pullback of an $\mathcal{M}$-map along any map should again be an $\mathcal{M}$-map; and

(d) the composite of any two $\mathcal{M}$-maps should also be one.

We consider the last property in Section 4, but we can already prove the first three:

Lemma 2.4 If Σ has a point $\star : \mathbf{1} \to \Sigma$ then $\mathsf{id}_A : A \rightarrowtail A$ (or any isomorphism $A' \cong A$) is the partial product of $\star, \star : A \times \mathbf{1} \rightrightarrows \Sigma$. $\square$

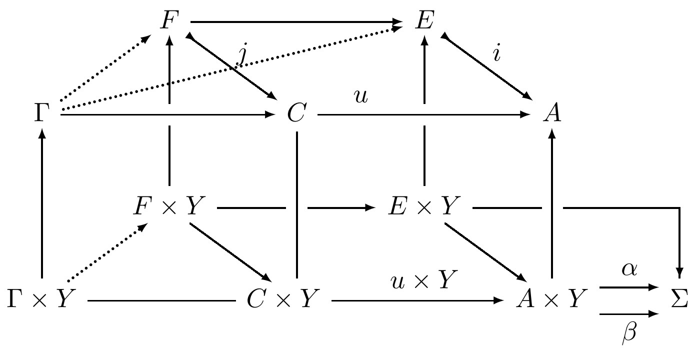
Lemma 2.5 Any subspace map $E \to A$ is mono.

Proof Let $e, e' : \Gamma \rightrightarrows E$ be such that $a \equiv i \cdot e$ is the same as $i \cdot e'$. Then the morphisms

$$\alpha \cdot (a \times Y), \quad \alpha \cdot (i \cdot e \times Y), \quad \alpha \cdot (i \times Y) \cdot (e \times Y),$$
$$\beta \cdot (i \times Y) \cdot (e \times Y) \quad \text{and} \quad \beta \cdot (a \times Y)$$

are the same, so a satisfies the pre-condition required for the universal property. Hence there is a *unique* $e : \Gamma \to E$ for which $i \cdot e$ is that same as a, which means that e' is the same as e. $\square$

Lemma 2.6 Partial product diagrams are stable under pullback.



Proof Suppose that $i : E \rightarrowtail A$ is the partial product of $\alpha, \beta : A \times Y \rightrightarrows \Sigma$ and let $u : C \to A$ be any map.

First suppose that the parellelogram at the top is a pullback. Given a map $\Gamma \to C$ with equal composites to Σ, there is a map $\Gamma \to E$ since E is a partial product and then $\Gamma \to F$ since F is a pullback.

Conversely, if F is formed as a partial product and we have maps $\Gamma \to C$ and $\Gamma \to E$ commuting at A then their composites to Γ are equal and the mediator $\Gamma \to F$ is given by the universal property of F as a partial product. Hence F is also a pullback. $\square$

Such pullbacks of subspaces are traditionally known as ***inverse images***. More abstractly, this result makes $\mathcal{M}$ a ***class of display maps*** [Tay99, Def. 8.3.2] that admits universal quantification along binary product projections. Syntactically, it will allow substitution for the free variable of type A in α and β.

Remark 2.7 Let i_1 and i_2 be the partial products of $A \times Y_1 \rightrightarrows \Sigma$ and $A \times Y_2 \rightrightarrows \Sigma$. Then the ***intersection*** $i_1 \cap i_2$ is the partial product of $A \times (Y_1 + Y_2) \rightrightarrows \Sigma$. That is, so long as the coproduct $Y_1 + Y_2$ exists in $\mathcal{Q}$ and $A \times (-)$ distributes over it. However, instead of relying on this, we shall treat intersection alongside partial product as a basic operation on subspaces.

If pullbacks are inverse images, what is the *direct image* for this notion of subspace?

Definition 2.8 A map $e : \Delta \to \Gamma$ is **Σ-*epi*** if, for any object Y and maps $\phi, \psi : \Gamma \times Y \rightrightarrows \Sigma$ for which the composites $\phi \cdot (e \times Y)$ and $\psi \cdot (e \times Y)$ are the same then ϕ and ψ were already the same.

Many roles have already been found in classical and synthetic domain theory [Hyl91, Tay91] for Σ-epis that are not surjective. The reason why we reject presheaf and PER models as "set theory" is that we intend to take the class of Σ-epis seriously and develop an "existential quantifier" $\ni$ for them in [BB].

Proposition 2.9 The class $\mathcal{E}$ of Σ-epis is ***orthogonal*** to that of partial product inclusions ($\mathcal{M}$). This means that, in any commutative trapezium (on the top face of the cube, $i \cdot f$ and $g \cdot e$ are the same), there is a unique fill-in $v : \Gamma \to E$ that makes the two triangles commute.

$$\begin{tikzcd}
\Delta \arrow[rr, "f"] \arrow[dr, "e"] & & E \arrow[dr, "i"] & & \\
& \Gamma \arrow[ur, dotted, "v"] \arrow[rr, "g"] & & A & \\
\Delta \times Y \arrow[uu, "\pi_0"] \arrow[rr, "f \times Y"] \arrow[dr, "e \times Y"] & & E \times Y \arrow[uu] \arrow[dr, "i \times Y"] \arrow[drr] & & \\
& \Gamma \times Y \arrow[uu, "\pi_0"] \arrow[ur, dotted, "v \times Y"] \arrow[rr, "g \times Y"] & & A \times Y \arrow[uu, "\pi_0"] \arrow[r, shift left, "\alpha"] \arrow[r, shift right, "\beta"'] & \Sigma
\end{tikzcd}$$

Proof Since the trapezium commutes and we have a partial product diagram, all paths from $\Delta \times Y$ to Σ have the same composite. Then, since e is Σ-epi, the composites $\Gamma \times Y \rightrightarrows \Sigma$ are also the same. Hence we may invoke the universal property of the partial product, which provides a unique v for which $i \cdot v$ is the same as g. Also, $v \cdot e$ is the same as f because i is mono. $\square$

Remark 2.10 Whilst epis in a *cartesian closed* category are always stable under product, beware that a map may be epi in a *sub*category without being epi in an *enclosing* category. Conversely, we are about to show how partial products capture equalisers of exponentials, so the class $\mathcal{M}$ consists of regular monos in the enclosing CCC, but these need not be expressible as equalisers in the subcategory.

The existence of partial products is not sufficient to make the class of Σ-epis stable under product, but this will follow from the other axioms that we shall add to definition of an equideductive category in Section 4. We shall find in [BB] that Σ-epis in such a category are the same as epis in the usual sense and that every morphism factorises as a Σ-epi followed by a partial product inclusion. In other words, these two classes form a factorisation system, in which orthogonality is the universal property.

Now we can return to partial products and enclosing CCCs, *cf.* Definition 4.3.

Lemma 2.11 Suppose that the exponential Σ^Y exists. Then the map $E \to A$ is the partial product of $\alpha, \beta : A \times Y \rightrightarrows \Sigma$ iff it is the equaliser of their exponential transposes $A \rightrightarrows \Sigma^Y$.

$$\begin{tikzcd}
 & E \arrow[dr, "i"] & & \\
\Gamma \arrow[ur, dotted, "e"] \arrow[rr, "a"] & & A \arrow[r, shift left, "\tilde{\alpha}"] \arrow[r, shift right, "\tilde{\beta}"'] & \Sigma^Y \\
 & E \times Y \arrow[uu] \arrow[dr, "i \times Y"] \arrow[ddr] & & \\
\Gamma \times Y \arrow[uu] \arrow[ur, dotted] \arrow[rr, "a \times Y"] \arrow[drr, "\alpha(a,y) = \beta(a,y)"'] & & A \times Y \arrow[uu] \arrow[r, shift left, "\tilde{\alpha} \times Y"] \arrow[r, shift right, "\tilde{\beta} \times Y"'] \arrow[d, shift right, "\alpha"'] \arrow[d, shift left, "\beta"] & \Sigma^Y \times Y \arrow[uu] \arrow[dl, "\mathsf{ev}"] \\
 & & \Sigma &
\end{tikzcd}$$

Proof By the defining universal property of the exponential Σ^Y, the transpose of the common composite $\Gamma \times Y \to A \times Y \rightrightarrows \Sigma$ in the lower triangle is the common composite $\Gamma \to A \rightrightarrows \Sigma^Y$ in the upper part of the diagram, but this tests the equaliser E. $\square$

Notice that ev played no role in this proof. We do not need an enclosing cartesian closed category or the right-hand column of the diagram but can instead concentrate on the partial products in a subcategory.

In order to show the converse, *i.e.* that partial products may always be seen as equalisers of exponentials in some enclosing cartesian closed category, we use the Yoneda embedding into a presheaf category. This takes us into the traditional set-theoretic world, where we remain until the end of the next section, but after that the remainder of the paper avoids it. The construction of an enclosing cartesian closed category that we give in [**?**] is finitary and does not use set theory.

Lemma 2.12 Let $\mathcal{Q}$ be a small category with finite limits and partial products and $\mathcal{S} \equiv \mathbf{Set}^{\mathcal{Q}^{\mathsf{op}}}$ be its topos of presheaves. Then the ***Yoneda embedding*** $\mathcal{Q} \rightarrowtail \mathcal{S}$, defined by $Y \mapsto \mathcal{Q}(-, Y)$, takes partial products to equalisers of exponentials.

Proof This functor is full and faithful and preserves products [Mac71] [Tay99, Thm. 4.8.12]. The value of the exponential presheaf Σ^Y at $\Gamma \in \mathcal{Q}$ must be

$$\begin{aligned} \Sigma^Y(\Gamma) \;\cong\; \mathcal{Q}(-, \Gamma) \to \Sigma^Y \;&\cong\; \mathcal{Q}(-, \Gamma) \times \mathcal{Q}(-, Y) \to \mathcal{Q}(-, \Sigma) \\ &\cong\; \Gamma \times Y \to \Sigma \;\equiv\; \mathcal{Q}(\Gamma \times Y, \Sigma), \end{aligned}$$

so $\Sigma^Y \equiv \mathcal{Q}(- \times Y, \Sigma)$. Then maps $\alpha : A \times Y \to \Sigma$ in $\mathcal{Q}$ correspond bijectively to natural transformations $\bar{\alpha} : \mathcal{Q}(-, A) \to \mathcal{Q}(- \times Y, \Sigma) \equiv \Sigma^Y$. This correspondence is itself natural in A, *i.e.* with respect to $a : \Gamma \to A$, so

$$\alpha \cdot (a \times Y) = \beta \cdot (a \times Y) \iff \overline{\alpha \cdot (a \times Y)} = \overline{\beta \cdot (a \times Y)} \iff \bar{\alpha} \cdot a = \bar{\beta} \cdot a.$$

For E to be the partial product in $\mathcal{Q}$ means that the first of these equations provides a unique e with $a = i \cdot i$, whilst for E to be the equaliser in $\mathcal{S}$ means that the last does so. $\square$

We can also characterise Σ-epis in the same terms:

Lemma 2.13 A map $e : \Delta \to \Gamma$ in $\mathcal{Q}$ is Σ-epi iff the Yoneda embedding takes it to a map $\bar{e}$ for which $\Sigma^{\bar{e}}$ is mono.

Proof Definition 2.8 says that the action on hom-sets

$$\mathcal{Q}(\Gamma \times (-), \Sigma) \longrightarrow \mathcal{Q}(\Delta \times (-), \Sigma)$$

given by pre-composition with a Σ-epi e is injective. $\square$

The next result is a useful tool for finding partial products in some semantic categories.

Lemma 2.14 Let $\mathcal{Q}$ be any category with finite limits such that each object $Y \in \mathcal{Q}$ has some object $SY \in \mathcal{Q}$ and a natural 1–1 function

$$\mathcal{Q}(- \times Y, \Sigma) \rightarrowtail \mathcal{Q}(-, SY).$$

Then $\mathcal{Q}$ has partial products and their inclusions are regular monos.

Proof The required equaliser in the presheaf category $\mathbf{Set}^{\mathcal{Q}^{\mathrm{op}}}$ is

$$\begin{tikzcd} E \arrow[r, tail] & \mathcal{Q}(-, A) \arrow[r, shift left] \arrow[r, shift right] & \mathcal{Q}(- \times Y, \Sigma) \arrow[r, tail] & \mathcal{Q}(-, SY). \end{tikzcd}$$

Since the equaliser $E' \rightarrowtail A \rightrightarrows SY$ exists in $\mathcal{Q}$ and the Yoneda embedding preserves it, we have $E \cong \mathcal{Q}(-, E')$, so E' is the partial product in $\mathcal{Q}$. $\square$

Remark 2.15 The natural transformation above is an isomorphism iff SY is the exponential Σ^Y in $\mathcal{Q}$ (Definition 4.3), in which case the inverse is given by composition with $\mathsf{ev} : SY \times Y \to \Sigma$. In an alternative approach to constructing cartesian closed extensions, Aurelio Carboni and Giuseppe Rosolini [CR00] define a ***weak exponential*** W of Y and Σ to be a map $e : W \times Y \to \Sigma$ for which composition with e defines a natural transformation in the *opposite* direction,

$$\mathcal{Q}(- \times Y, \Sigma) \twoheadleftarrow \mathcal{Q}(-, W),$$

that is componentwise *surjective*.

Another thing that we can do with categories embedded in a set-theoretic world is to form the isomorphism classes of subobjects and then collect these into a semilattice.

Notation 2.16 For any object $Y \in \mathcal{Q}$, let $\mathsf{Sub}(Y) \equiv \mathcal{M}/Y$ be the class of isomorphism classes of $\mathcal{M}$-maps into Y. By Lemma 2.6, the pullback of an $\mathcal{M}$-map along any $f : Z \to Y$ is another $\mathcal{M}$-map, so we have a functor with $\mathsf{Sub}(f) \equiv f^* : \mathsf{Sub}(Y) \to \mathsf{Sub}(Z)$. In the case $f \equiv \pi_0 : Y \times X \to Y$, f^* has a right adjoint iff this is the partial product $X \Rrightarrow$. Lemma 2.6 also says that this partial product satisfies a Beck–Chevalley condition:

$$\begin{tikzcd} X \times Y' \arrow[r, "\pi_1"] \arrow[d, "\mathsf{id} \times f"'] \arrow[dr, phantom, "\lrcorner", very near start] & Y' \arrow[d, "f"] \\ X \times Y \arrow[r, "\pi_1"] & Y \end{tikzcd} \qquad \begin{tikzcd} \mathsf{Sub}(X \times Y') \arrow[r, shift left=2, "X \Rrightarrow"] \arrow[r, phantom, "\top"] & \mathsf{Sub}(Y') \arrow[l, shift left=2, "\times X"] \\ \mathsf{Sub}(X \times Y) \arrow[u, "(\mathsf{id} \times f)^*"] \arrow[r, shift left=2, "X \Rrightarrow"] \arrow[r, phantom, "\top"] & \mathsf{Sub}(Y) \arrow[l, shift left=2, "\times X"] \arrow[u, "f^*"'] \end{tikzcd}$$

We shall see in [BB] that $\mathsf{Sub}(A)$ is a lattice with $\top$, $\bot$, $\&$ and $\curlyvee$, whilst $\mathsf{Sub}(\mathbf{0}) = \mathbf{1}$ and $\mathsf{Sub}(X+Y) = \mathsf{Sub}(X) \times \mathsf{Sub}(Y)$.

This setting also leads to partial products:

Proposition 2.17 Let $\mathcal{Q}$ be a category in which
(a) all finite limits exist;
(b) every map factorises as an epi followed by a regular mono;
(c) products preserve epis;
(d) considered as a semilattice map $\mathsf{Sub}(X) \to \mathsf{Sub}(X \times Y)$ between the classes of regular monos into X and $X \times Y$, the product $(-) \times Y$ has a right adjoint.

Then $\mathcal{Q}$ has partial products. $\square$

Semilattice completeness then provides the required adjoint:

Corollary 2.18 Let $\mathcal{Q}$ be a complete and regularly well powered category in which regular monos compose and products preserve epis and unions of subobjects. Then $\mathcal{Q}$ has partial products.

Proof These conditions say that $\mathsf{Sub}(X)$ is a set (rather than a class) and carries the structure of a complete lattice, for which $(-) \times Y$ is a homomorphism, so it has a right adjoint. $\square$

3 Examples

Now we shall construct partial products in the categories of sets and of sober topological spaces and also in a model defined from recursion theory. The object Ω or Σ in these three categories is injective, whilst other sets, spaces and recursive types can be represented using them. These are the properties that we shall put together to give the definition of an equideductive category in the next section.

The category of all locales does not admit partial products [**?**], but the fact that all *countably presented* locales are spatial suggests that these may form another equideductive category. However, the notion of "countability" that is needed for this seems to be neither the classical one nor recursive enumerability.

Proposition 3.1 Any elementary topos $\mathcal{Q}$ with $\Sigma \equiv \Omega$ has partial products.

Proof Any topos has all finite limits and exponentials. Hence it has partial products by Lemma 2.11, but it is worth spelling this out as an application of Lemma 2.14. Any morphism $A \times Y \to \Sigma$ classifies a subobject of the rectangle $A \times Y$, so it may also be seen as a binary relation $A \rightharpoondown Y$ or as a function $\bar{\alpha} : A \to \mathcal{P}(Y)$ by

$$\bar{\alpha}x \;\equiv\; \{y \mid \alpha xy\}, \quad \text{so} \quad y \in \bar{\alpha}x \iff \alpha xy.$$

Moreover, this correspondence is 1–1: $\alpha = \beta : A \times Y \to \Sigma$ iff $\bar{\alpha} = \bar{\beta} : A \to \mathcal{P}(Y)$, because this is what equality of subsets means. (It is surjective too, but this is not significant and will not happen in the other examples.) It is also natural, *i.e.* it respects precomposition with any map $a : \Gamma \to A$:

$$\alpha(az) \;=\; \{y \mid \alpha(az)y\} \;=\; \{y \mid (\alpha \cdot (a \times \mathsf{id}))zy\},$$

so Lemma 2.14 gives the partial product.

Remark 3.2 In Example 2.2 we saw how partial products capture *external* intersection of families of subobjects. In a topos, suppose that the subobject $V \subset A \times Y$ is classified by $\alpha : A \times Y \to \Sigma$, so V is the equaliser of α and $\beta \equiv \top$. Then the intersection $E \subset A$ is classified by $\forall_Y \alpha \equiv \forall y \in Y.\, \big(\alpha(a,y) = \top\big)$.

If $\Gamma \rightarrowtail A$ is classified by ψ then $\psi \cdot \pi_1 \leqslant \alpha$ whilst $\psi \leqslant \forall_Y \alpha$. Hence $(-) \cdot \pi_1 \dashv \forall_Y(-)$.

In the Proposition, since

$$\bar{\alpha}x = \bar{\beta}x \iff \forall y{:}Y.\, (y \in \bar{\alpha}x \Leftrightarrow y \in \bar{\beta}x) \iff \forall y{:}Y.\, (\alpha xy \Leftrightarrow \beta xy),$$

the quantifier $\forall$ has its usual logical meaning for the set Y.

Proposition 3.3 The object Ω and in general any powerset $\mathcal{P}(A) \equiv \Omega^A$ in a topos is injective.

Proof Any subsubset is a subset. $\square$

Proposition 3.4 The full subcategory of a topos whose objects are powersets is closed under finite products and exponentials $\Omega^{(-)}$. $\square$

Proposition 3.5 Any topos has enough injectives, in the sense that any object X may be expressed as an equaliser of powersets,

$$X \rightarrowtail \Omega^A \rightrightarrows \Omega^B. \qquad \square$$

We can apply similar arguments to the traditional category **Sp** of topological spaces and continuous functions, now taking Σ to be the Sierpiński space.

Proposition 3.6 The category **Sp** has partial products.

Proof Its maps $X \to \Sigma$ correspond to open subspaces of X and we write $\mathcal{O}(X)$ for the lattice of them. Hence maps $\alpha : A \times Y \to \Sigma$ classify open subspaces $W \subset A \times Y$ in the *Tychonov topology* on the product of the underlying sets. Following the previous example, we define a natural 1–1 function $|A| \to \mathcal{O}(Y)$ by

$$\bar{\alpha}x \;\equiv\; \{y : Y \mid \alpha xy\} \;=\; \bigcup\nolimits^{\uparrow}\{V \in \mathcal{O}(Y) \mid \exists U \in \mathcal{O}(A).\, x \in U \,\&\, U \times V \subset W\},$$

which yields an *open* subset because W is a union of open rectangles like $U \times V$. (To see that this union is directed, consider $U \times V \equiv A \times \emptyset \subset W$ and $(U_1 \cap U_2) \times (V_1 \cup V_2) \subset W$.)

In order to make $\bar{\alpha}$ into a morphism of $\mathcal{Q}$ (continuous function), we need to assign a topology to the set $\mathcal{O}(Y)$. There are several ways of doing this, but unlike $\mathcal{P}(Y)$ in **Set**, the result only obeys the universal property of the exponential Σ^Y (Definition 4.3) when Y is *locally compact*. However, in order to apply Lemma 2.14, we only need $\bar{\alpha} : A \to \mathcal{O}(Y)$ to be an $\mathcal{Q}$-map (continuous function) that is *mono* and natural in A.

If $\mathcal{V} \subset \mathcal{O}(Y)$ is a ***Scott open*** family of open subspaces then

$$\begin{aligned} \bar{\alpha}^{-1}(\mathcal{V}) &= \big\{x \in A \,\big|\, \bigcup\nolimits^{\uparrow}\{V \in \mathcal{O}(Y) \mid \exists U \in \mathcal{O}(A).\, x \in U \,\&\, U \times V \subset W\} \in \mathcal{V}\big\} \\ &= \{x \in A \mid \exists V \in \mathcal{V}.\, \exists U \in \mathcal{O}(A).\, x \in U \,\&\, U \times V \subset W\} \\ &= \bigcup \{U \in \mathcal{O}(A) \mid \exists V \in \mathcal{V}.\, U \times V \subset W\} \;\subset\; A \end{aligned}$$

is open too, so $\bar{\alpha}$ is continuous. It is mono and natural because this is the case for points, so once again $\forall$ is the ordinary universal quantifier. $\square$

Proposition 3.7 The Sierpiński space Σ and more generally any continuous lattice with the Scott topology is injective with respect to subspace inclusions [Sco72]. $\square$

Proposition 3.8 The full subcategory of algebraic lattices with the Scott topology is closed under finite products and exponentials $\Sigma^{(-)}$. $\square$

However, we have to modify our choice of category:

Proposition 3.9 A space X may be expressed as an equaliser of algebraic lattices,

$$X \rightarrowtail \Sigma^A \rightrightarrows \Sigma^B,$$

iff it is *sober* (in the traditional sense, rather than our abstract one in Section 7).

Proof The partial product is sober because it is given by an equaliser of sober spaces, which form a reflective subcategory [Joh82, Cor. II 1.7(ii)]. $\square$

Remark 3.10 **Sob** satisfies the previous results.

Remark 3.11 Our third example comes from recursion theory. It is a simplification of the construction of a cartesian closed category of partial equivalence relations (PERs) on a partial combinatory algebra. Indeed, the notion of equideductive category was motivated by the fact that the construction of equalisers in PER models leaves the quotient part of the PER untouched.

In order to emphasise the role of quantifier complexity, we would like to work with $\mathbb{N}$ rather than some less familiar recursive–topological object. However, it is much more convenient to use the set $\mathbb{T}$ of binary trees because $\mathbb{T}\times\mathbb{T}+\mathbf{1}\cong\mathbb{T}\cong\mathsf{List}\,(\mathbb{T})$, with the usual notation $[h|t]$ for pairs and $[a,b,c,\ldots]$ for lists. The lattice R of recursively enumerable subsets of $\mathbb{T}$ will have to be an object in our model, and we represent it as the quotient of $\mathbb{T}$ by a single equivalence relation, $\sim$. However, after that, we only need PERs that are subrelations of $\sim$.

We rely on the existing literature to provide the details of the categorical structure.

Notation 3.12 Stephen Kleene [Kle43] showed that there is a decidable primitive recursive predicate T with the following property: Any recursively enumerable or Σ^0_1 predicate ϕ on $\mathbb{T}$ has a ***Kleene normal form***, *i.e.* some number ("program") p such that

$$\phi(x) \iff \bar{p}(x) \;\equiv\; \exists h.\, T[p,x|h].$$

The predicate T and codes p are derived structurally from whatever programming language we use to define computation. We introduce special programs called *combinators* as needed to do this; they usually just re-arrange branches of the tree.

Inclusion between RE subsets is represented by a preorder relation between their representing programs,

$$(p\preccurlyeq q) \;\equiv\; \big(\forall x.\, \bar{p}(x)\Rightarrow\bar{q}(x)\big) \;\equiv\; \forall xh.\,\exists k.\,\neg T[p,x|h]\vee T[q,x|k],$$

which is a Π^0_2-formula in $p,q:\mathbb{T}$. These codes represent the same RE subset iff

$$(p\sim q) \;\equiv\; (p\preccurlyeq q)\;\wedge\;(q\preccurlyeq p).$$

The lattice R of RE subsets of $\mathbb{T}$ is then the quotient poset of the preorder $(\mathbb{T},\preccurlyeq)$.

A recursive function $R\to R$ is by definition a recursive function $F:\mathbb{T}\to\mathbb{T}$ that respects $\sim$. Then by Kleene's theorem there is a program p with

$$\exists h.\, T[Fx,y|h] \iff \exists k.\, T[p,x,y|k].$$

Conversely, the (total primitive) recursive function $Gx\equiv[\alpha,p,x]$ gives rise to p as above, where we define a combinator α so that $T\big[[\alpha,p,x],y|h\big]\equiv T[p,x,y|h]$, and then $F\sim G$. Hence we may use either representation.

Definition 3.13 The ***crude recursive model*** is the category of subsets of R and total computable functions between them that respect $\sim$.

More precisely, an *object* of this category is a context consisting of a finite set $\vec{x}$ of variables (of type $\mathbb{T}$) and a first order predicate $\mathfrak{p}(\vec{x})$. We shall be interested in the ***quantifier complexity*** this predicate, *i.e.* the number of alternations of $\forall$ and $\exists$, these being allowed to range over $\mathbb{T}$.

A *morphism* $\vec{f} : [\vec{x}, \mathfrak{p}(\vec{x})] \longrightarrow [\vec{y}, \mathfrak{q}(\vec{y})]$ is an equivalence class of sequences of computable functions $f_j(\vec{x})$, one for each variable y_j, such that

$$\begin{array}{lcl} \vec{x},\ \mathfrak{p}(\vec{x}) & \vdash & \vec{f}(\vec{x})\downarrow \quad\text{and}\quad \mathfrak{q}\big(\vec{f}(\vec{x})\big) \\ \vec{x},\ \vec{x}',\ \vec{f}(\vec{x})\downarrow,\ \vec{x} \preccurlyeq \vec{x}' & \vdash & \vec{f}(\vec{x}')\downarrow \quad\text{and}\quad \vec{f}(\vec{x}) \preccurlyeq \vec{f}(\vec{x}'), \end{array}$$

where $\vec{f}$ and $\vec{g}$ define the same morphism if $\quad \forall\vec{x}.\, \mathfrak{p}(\vec{x}) \Rightarrow \vec{f}(\vec{x}) \sim \vec{g}(\vec{x})$.

If the morphism is represented by identity maps then it is called a ***crude inclusion***. It is crude in the sense that it need may but need not be a partial product inclusion.

Lemma 3.14 The crude recursive model is a category in which composition is defined by substitution, whilst finite products are given by concatenation of the lists of variables and conjunction of the predicates. $\square$

Lemma 3.15 The object $R \equiv [x, \top]$ is injective with respect to crude inclusions. Moreover every object has a crude inclusion into R. $\square$

Lemma 3.16 $\Sigma \lhd R$ and $\Sigma^R \lhd R$, also $\mathbb{N}$ and $R = \Sigma^{\mathbb{N}}$. $\square$

Lemma 3.17 Equalisers are given by

$$\begin{tikzcd} \big[\vec{x},\ \mathfrak{p}(\vec{x}) \,\&\, \big(\vec{f}(\vec{x}) \sim \vec{g}(\vec{x})\big)\big] \arrow[r, tail] & {[\vec{x},\ \mathfrak{p}(\vec{x})]} \arrow[r, shift left, "\vec{f}"] \arrow[r, shift right, "\vec{g}"'] & {[\vec{y},\ \mathfrak{q}(\vec{y})]} \arrow[r, tail] & {[\vec{y}]} \end{tikzcd}$$

and so have the same quantifier complexity as the given objects. $\square$

Lemma 3.18 Partial products are given by diagrams of the form

$$\begin{tikzcd} & E \equiv [\vec{x},\ \mathfrak{p}(\vec{x}) \,\&\, \mathfrak{r}(\vec{x})] \arrow[dr, "\vec{\imath}"] & & Y \equiv [\vec{y},\ \mathfrak{q}(\vec{y})] \\ \Gamma \arrow[ur, dotted, "\vec{e}"] \arrow[rr, "\vec{a}"] & & A \equiv [\vec{x},\ \mathfrak{p}(\vec{x})] & \\ & E \times Y \arrow[uu] \arrow[dr, "\vec{\imath} \times Y"] \arrow[drr] & & \\ \Gamma \times Y \arrow[uu, "\pi_0"] \arrow[ur, dotted, "\vec{e} \times Y"] \arrow[rr, "\vec{a} \times Y"] & & A \times Y \arrow[uu, "\pi_0"'] \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & \Sigma \lhd R \end{tikzcd}$$

where $\mathfrak{r}(\vec{x}) \ \equiv\ \forall\vec{y}.\, \mathfrak{q}(\vec{y}) \Rightarrow f(\vec{x},\vec{y}) \sim g(\vec{x},\vec{y})$.

Proof With $\mathfrak{r}(\vec{x})$ given in this way, the partial product obeys the rule

$$\frac{\Gamma \vdash \vec{a}, \quad \mathfrak{p}(\vec{a}) \qquad \Gamma,\ \vec{y},\ \mathfrak{q}(\vec{y}) \vdash f\vec{a} \sim g\vec{a}}{\Gamma \vdash \mathfrak{p}(\vec{a})\ \&\ \mathfrak{r}(\vec{a}).}$$

$\square$

Corollary 3.19 The quantifier complexity of the characteristic predicates of
(a) retracts of R^n is decidable;
(b) open subspaces of R is at most Σ_1^0;
(c) finite limits of copies of R is at most Π_2^0;
(d) partial products of R is at most Π_3^0; and
(e) partial products nested k deep is at most Π_{k+2}^0.
Therefore not all partial product inclusions (the class $\mathcal{M}$) need be representable as equalisers, notwithstanding the factorisation of general maps into (product-stable) epis and partial product inclusions. $\square$

We also give a sketch of the proof of the existence of the necessary exponentials.

Proposition 3.20 The exponential R^R exists in the crude recursive model and has $R^R \lhd R$.

Proof We have to define a natural retraction

$$(f : \Gamma \times R \to R) \quad \lhd \quad (g : \Gamma \to R).$$

It is convenient to represent morphisms by programs for recursively enumerable predicates as above and we introduce combinators λ and @ such that

$$T[f, \gamma, p, x|h] \iff T\big[[\lambda, f|\gamma], [p, x]|h\big] \iff T\big[@, [\lambda, f|\gamma], p, x|h\big]. \qquad \square$$

Definition 3.21 The ***recursive model*** is the full subcategory consisting of objects that are definable from R using partial products. We leave it to the interested reader to characterise those first order predicates that arise in this way and find examples that do not or which require more than a specified depth of nesting.

Theorem 3.22 The recursive model is an equideductive category. $\square$

4 Equideductive categories

The definition of an equideductive category that we shall now give was motivated by the leading examples in the previous section. In particular, although the topological and recursive model do not have general function-spaces, they do have a full subcategory $\mathcal{A}$ of objects A for which the exponential Σ^A exists and is injective. In the topological case, such spaces A are locally compact and their exponentials Σ^A are continuous distributive lattices with the Scott topology.

Remark 4.1 We showed in Section 2 how partial product inclusions ($\mathcal{M}$-maps) can be considered as "subspaces", but we still have to show that they compose. Considering the partial product diagram (Definition 2.1) that gives rise to the inclusion $F \rightarrowtail E$, we could prove this by *lifting* the defining maps like this:

$$\begin{tikzcd} E \times B \arrow[rr, tail] \arrow[dr, "\alpha"'] & & A \times B \arrow[dl, dotted] \\ & \Sigma & \end{tikzcd}$$

As a special case of Lemma 2.6, if $E \rightarrowtail A$ is in $\mathcal{M}$ then so is $E \times B \rightarrowtail A \times B$ for any object B. We what we need therefore is

Axiom 4.2 The object Σ is ***injective*** with respect to $\mathcal{M}$-maps, *i.e.* it has the lifting property above.

If Σ is injective then so too are its exponentials Σ^A. But, before proving this, we state the definition because we shall need to refer to it quite frequently.

Definition 4.3 The object Σ^A is the ***exponential*** of A, if for any map $\sigma : \Gamma \times A \to \Sigma$ there is a unique ***exponential transpose*** $\tilde{\sigma} : \Gamma \to \Sigma^A$ such that $\mathsf{ev} \cdot (\tilde{\sigma} \times A)$ is the same as σ:

$$\begin{tikzcd}
\Delta \times A \arrow[r, "u \times A"] & \Gamma \times A \arrow[rr, "\sigma"] \arrow[dr, dotted, "\tilde{\sigma} \times A"'] & & \Sigma \\
& & \Sigma^A \times A \arrow[ur, "\mathsf{ev}"'] &
\end{tikzcd}$$

We shall call the object A ***exponentiable*** if such an exponential Σ^A exists. It will be important in the definition of an equideductive category that *every* object Γ respect the universal property of Σ^A, even if Γ is not itself exponentiable. The standard use of the term exponentiable requires Y^A for every object Y of the category; we construct this in an equideductive category in Proposition **??**.

It is also convenient to write ev' for ev with its arguments switched; then the transpose of ev' is called η.

Lemma 4.4 Transposition is *natural* with respect to precomposition with $u : \Gamma \to \Delta$ in the sense that

$$\text{the transpose of} \quad \sigma \cdot (u \times A) \quad \text{is} \quad \tilde{\sigma} \cdot u. \qquad \square$$

Lemma 4.5 Let B be any exponentiable object. Then Σ^B is also injective with respect to $\mathcal{M}$.

$$\begin{tikzcd}
E \arrow[rr, tail] \arrow[dr, "\tilde{\alpha}"'] & & A \arrow[dl, dotted] \\
& \Sigma^B &
\end{tikzcd}$$

Proof Exponential transposition takes the triangle with vertex Σ^B into the one in Remark 4.1. Since the class $\mathcal{M}$ is closed under product with any object B and Σ is injective with respect to $\mathcal{M}$-maps, there is a map $A \times B \to \Sigma$ and so one $A \to \Sigma^B$ [Joh82, Lemma VII 4.10]. $\square$

In order to make use of the subcategory $\mathcal{A}$ to work with the whole of $\mathcal{Q}$ we need

Axiom 4.6

There are ***enough injectives***: each $\mathcal{Q}$-object has an $\mathcal{M}$-map $X \rightarrowtail \Sigma^B$ for some exponentiable object B.

This is enough as it stands in **Set** and **Sob**, but the recursive and abstract models are more complicated. We saw in the recursive one that this complexity agrees with that of the quantifiers in the characteristic predicates. In general, $X \rightarrowtail \Sigma^B$ may the inclusion of a partial product defined by a parallel pair $\Sigma^B \times Y \rightrightarrows \Sigma$ for which Y need not be an urspace.

We therefore require that each object X be expressible, not just using a single $\mathcal{M}$-map, but by a finite sequence of such steps.

Recall from Lemma 2.5 that $\mathcal{M}$ is contained in the class of all *monos*, but it also contains all *regular* monos:

Lemma 4.7 Every equaliser in $\mathcal{Q}$ is a partial product, *i.e.* the class $\mathcal{M}$ contains all regular monos.

Proof The equaliser $E \rightarrowtail X \rightrightarrows Y \rightarrowtail \Sigma^A$ is targeted at an exponential, so it is a partial product. $\square$

In the traditional concrete category described in the previous section, Lemma 2.14 applies: any $\mathcal{M}$-map can conversely be expressed as equaliser $X \rightarrowtail \Sigma^B \rightrightarrows \Sigma^C$ where both B and C belong to $\mathcal{A}$. In other words, $\mathcal{M}$ consists exactly of the *regular monos* in $\mathcal{Q}$.

Remark 4.8 Notice that we have tested the exponential Σ^B with maps from *arbitrary* objects E of the category $\mathcal{Q}$, not just from other As; this will become very important later in this paper.

We therefore ask for closure under exponentials as an axiom:

Definition 4.9 A ***class of urspaces*** is a full subcategory $\mathcal{A} \subset \mathcal{Q}$ of a category with finite limits and a chosen object Σ such that
(a) the objects $\mathbf{1}$ and Σ of $\mathcal{Q}$ belong to $\mathcal{A}$;
(b) $\mathcal{A} \subset \mathcal{Q}$ is closed under binary product; and
(c) $\mathcal{A}$ has exponentials $\Sigma^{(-)}$ (but not necessarily all B^A) and objects of $\mathcal{Q}$ respect them, as in Definition 4.3, where each Σ^A also belongs to $\mathcal{A}$.
There is a contrast between $\mathcal{Q}$ and $\mathcal{A}$ in that $\mathcal{Q}$ has equalisers but not necessarily exponentials, whilst $\mathcal{A}$ has exponentials Σ^A but not necessarily equalisers.

In this situation we get our first glimpse of how useful the new logic is, because it can say when two functions $\phi, \psi : A \rightrightarrows \Sigma$ are equal. Beware, however, that this ***extensionality property*** must be *interpreted formally in the logic*: it does not mean that there are enough morphisms $\mathbf{1} \to A$ to distinguish ϕ from ψ. For example, any topos has this structure, but need not be well pointed.

Lemma 4.10 Suppose that the exponential Σ^A exists in $\mathcal{Q}$ and is respected by all objects of $\mathcal{Q}$. Then the partial product $\forall y{:}A.\ (\phi y = \psi y)$ is the diagonal $(\phi = \psi)$ of $\Sigma^A \times \Sigma^A$.

$$\begin{tikzcd}
 & & \Sigma^A \arrow[dr, "\Delta"] & & \\
\Gamma \arrow[urr, dotted, "\tilde{f}"] \arrow[rrr, "(\phi,\psi)"'] & & & \Sigma^A \times \Sigma^A & \\
 & & \Sigma^A \times A \arrow[uu] \arrow[dr, "\Delta \times A"] \arrow[drr, "\mathsf{ev}"] & & \\
\Gamma \times A \arrow[uu, "\pi_0"] \arrow[urr, dotted, "\tilde{f} \times A"] \arrow[rrr, "(\phi,\psi) \times A"'] \arrow[rrrr, bend right, "f"'] & & & \Sigma^A \times \Sigma^A \times A \arrow[uu, "\pi_{01}"'] \arrow[r, shift left, "\mathsf{ev}\cdot\pi_{02}"] \arrow[r, shift right, "\mathsf{ev}\cdot\pi_{12}"'] & \Sigma
\end{tikzcd}$$

Proof We need to show that the diagram above is a partial product, with A, Σ^A and Σ^A in place of Y, A and E in Definition 2.1. If the lower triangle commutes then ϕ and ψ are both exponential transposes of the common composite f, so $\phi = \tilde{f} = \psi$. $\square$

Putting all of these assumptions on $\mathcal{Q}$ and $\mathcal{A}$ together, we make the

Definition 4.11 An ***equideductive category*** consists of
(a) a category $\mathcal{Q}$ with all finite limits;
(b) a pointed object $\star : \mathbf{1} \to \Sigma$ in $\mathcal{Q}$; and
(c) a full subcategory $\mathcal{A} \subset \mathcal{Q}$ of urspaces; such that
(d) $\mathcal{A} \subset \mathcal{Q}$ is closed under products;
(e) Σ and all powers Σ^A for $A \in \mathcal{A}$ exist in $\mathcal{Q}$ and belong to $\mathcal{A}$;
(f) $\mathcal{Q}$ has partial products based on the object Σ;
(g) Σ is injective with respect to all of the maps in the class $\mathcal{M}$; and
(h) there are enough injectives, in the "well founded" sense of Axiom 4.6, where
(i) $\mathcal{M}$ is the class of monos defined by the partial products and intersections; and
(j) all objects $\Gamma \in \mathcal{Q}$ respect the universal properties mentioned.
The point $\star$ was used in Lemma 2.4 and will be needed in [BB], but is not used again in this paper.

Examples 4.12 **Set**, any topos, **Sob** and the recursive model are equideductive categories but **Loc** is not.

Recall that, for the subcategory $\mathcal{A}$ of urspaces in each of these semantic models, there is a choice between
(a) *all* sets in **Set** or *all* locally compact spaces in **Sob**, and
(b) just the *full powersets* or just the *algebraic lattices* equipped with the Scott topology.
As we move towards consideration of the symbolic language, we shall find that the smaller class is more useful for computation, whilst the larger is better for expressing mathematics. We therefore consider the relationship between them.

Lemma 4.13 If the exponential Σ^X exists in an equideductive category $\mathcal{Q}$ then there is some urspace $A \in \mathcal{A}$ and maps $i : X \to A$ and $I : \Sigma^X \to \Sigma^A$ with $\Sigma^i \cdot I = \mathsf{id}_{\Sigma^X}$.

Proof Since there are enough injectives, there are an urspace B and a $\mathcal{M}$-map $J : \Sigma^X \rightarrowtail \Sigma^B$. But Σ^X is itself injective by Lemma 4.5, so there is some $P : \Sigma^B \twoheadrightarrow \Sigma^X$ with $P \cdot J = \mathsf{id}_{\Sigma^X}$. Then the transpose $i : X \to A \equiv \Sigma^{\Sigma^B}$ of P and $I \equiv J \cdot \eta_{\Sigma^A} : \Sigma^X \to \Sigma^A$ satisfy $\Sigma^i \cdot I = \mathsf{id}_{\Sigma^X}$. □

Proposition 4.14 The class of all exponentiable objects in an equideductive category provides another class of urspaces.

Proof Any retract of an exponentiable object is also exponentiable. Also, if Σ^X exists then so does Σ^{Σ^X}, because if X is a Σ-split subspace of A then Σ^X is a retract of Σ^A. □

Proposition 4.15 The class of all injective objects in an equideductive category provides a class of urspaces.

Proof An object is injective iff it is a retract of an exponential. Hence products and exponentials of injectives are injective.. □

5 The restricted lambda calculus

Now we begin to develop a symbolic language for the categorical structure that we have described. In this section we start with the exponentials, for which we only consider Σ^A and not general B^A. This is motivated by the importance of open subspaces in topology and the idea that termination is the fundamental observable property in computation, but also by the connection with injectives in Lemma 4.5.

We spell out this well known material in some detail because it serves as a plan for our treatment of the $\forall\!\!\Rrightarrow$ language for partial products later.

However, since equality in equideductive logic is an "interrogative" notion that is different from "factual" rewriting in the λ-calculus, we shall refer to (the symmetric transitive closure of) the latter as ***interchangeability*** and write $\leftrightarrow$ instead of $=$ for it.

The restricted λ-calculus will provide the arena for computation, whereas the richer calculus is that of proof. The latter will have more complicated types or contexts that are defined by predicates, and terms or morphisms represented by partial equivalence classes. For this reason we shall use the prefix "ur" (meaning original) for the underlying calculus.

Remark 5.1 Recall that products and exponentials are defined by the bijections

$$\frac{\Gamma \xrightarrow{a} A \qquad \Gamma \xrightarrow{b} B}{\Gamma \xrightarrow{\langle a,b\rangle} A\times B} \qquad \text{or} \qquad \frac{\Gamma \vdash a \leftrightarrow \pi_0 p : A \qquad \Gamma \vdash b \leftrightarrow \pi_1 p : B}{\Gamma \vdash p \leftrightarrow \langle a,b\rangle : A\times B}$$

and

$$\frac{\Gamma\times A \xrightarrow{\sigma} \Sigma}{\Gamma \xrightarrow{\tilde\sigma} \Sigma^A} \qquad \text{or} \qquad \frac{\Gamma,\ x:A \ \vdash\ \sigma \leftrightarrow \phi x \equiv \mathsf{ev}(\phi,x) : \Sigma}{\Gamma \ \vdash\ \phi \leftrightarrow \lambda x.\,\sigma : \Sigma^A}$$

where $\langle a,b\rangle$ is the unique p such that $a=\pi_0 p$ and $b=\pi_1 p$ and $\tilde\sigma$ is the transpose of σ (Definition 4.3), being unique such that $\sigma = \mathsf{ev}\cdot(\tilde\sigma\times A)$.

There are two different ways of forcing these correspondences to be bijective. In symbolic logic the usual approach is to state the the ***beta-*** and ***eta-laws***,

$$(\lambda x.\,\sigma)x \ \leftrightarrow\ \sigma \qquad \text{and} \qquad \lambda x.\,\phi x \ \leftrightarrow\ \phi.$$

$$\pi_0\langle a,b\rangle \ \leftrightarrow\ a, \quad \pi_1\langle a,b\rangle \ \leftrightarrow\ b \quad \text{and} \quad \langle \pi_0 p, \pi_1 p\rangle \ \leftrightarrow\ p,$$

and also require the operations to ***transmit interchangeability*** and ***substitute***.

In category theory, on the other hand, the product $A\times B$ is defined by saying that for any two maps a and b there is a *unique* map p that satisfies the β-rules $\pi_0\cdot p\leftrightarrow a$ and $\pi_1\cdot p\leftrightarrow b$. We refer to this uniqueness condition as the ***extensionality rule***, which can also be seen as saying that the maps π_0 and π_1 are ***jointly mono***.

In equideductive logic we intend to use extensionality rules for both connectives. This property for $\Sigma^{(-)}$ is given semantically by Lemma 4.10, but this uses partial products and so is outside the scope of this section. We are therefore obliged to use the $\lambda\eta$-rule here and prove that the two formulations are equivalent later. Since there is no difficulty with using extensionality to define $\times$, that is what we do in both calculi (Axiom 5.8).

Axiom 5.2 The anatomy of the ***restricted λ-calculus*** consists of

(a) *urtypes*, A, which are generated from base types such as $\mathbf{0}$, $\mathbf{1}$ and $\mathbb{N}$ by $\times$ and $\Sigma^{(-)}$ (which is written $(-)\to\Sigma$ elsewhere);

(b) *urcontexts*, Γ, which are just lists of distinct variables, to each of which is assigned an urtype (there are no equations yet);

(c) *urterms*, a, which are generated from variables, $\star$, $\langle\ ,\ \rangle$, π_0, π_1, λ, ev and operation symbols, according to the rules below;

(d) *urterm-formation judgements*, $\Gamma \vdash a : A$, which assert that the urterm a is well formed, only contains (freely) the variables in the urcontext Γ, and is of urtype A;

(e) *interchange judgements*, $\Gamma \vdash a \leftrightarrow b : A$, which say both that the urterms are well formed and that one may be replaced by the other; such judgements are generated from *particular interchanges* and the rules of the calculus.

Remark 5.3 We may assume that operation symbols and particular interchanges have base types or Σ as their result types. However, we need to generate part of the language in order to define their argument types and the urterms that may be interchanged. We write $\mathcal{L}$ for the collection of base types, operation symbols and particular interchanges of a ***particular language***. By adding these things, Proposition 6.6 adapts the pure calculus to match any given category with products and exponentials.

Axiom 5.4 There are ***structural rules*** that manipulate judgements. For the variables in the urcontext, these are:

(a) *identity*: any variable from the urcontext is a well formed urterm of its urtype;

(b) *weakening*: new urtyped variables may be added to the urcontext of any valid judgement of either kind;

(c) *exchange*: the variables in the urcontext may be permuted, so we regard the list as unordered;

(d) *contraction*: if two variables in an urcontext have the same urtype then one may be substituted for the other, which is deleted from the urcontext (as on the left below);

$$\frac{\Gamma,\ x, y : A\ \vdash\ b : B}{\Gamma,\ x : A\ \vdash\ [x/y]^* b : B} \qquad\qquad \frac{\Gamma\ \vdash\ a : A \qquad \Delta,\ x : A\ \vdash\ b : B}{\Gamma,\ \Delta\ \vdash\ [a/x]^* b : B}$$

(e) *cut* (on the right above), in which Γ, Δ is the union of the lists (with repetitions removed) and $[a/x]^* b$ denotes *substitution* of the urterm a for the variable x in the urterm b, avoiding *capture* by λ-abstraction. We explain why we use a star in the notation for substitution in Section 11.

Axiom 5.5 There are also structural rules for manipulating interchanges:

(a) they are reflexive, symmetric and transitive;

(b) they admit weakening, exchange and contraction of the variables in the urcontext; and

(c) they admit cut or substitution, both of an urterm into an interchange and *vice versa*:

$$\frac{\Gamma\ \vdash\ a : A \qquad \Delta,\ x : A\ \vdash\ c \leftrightarrow d : C}{\Gamma,\ \Delta\ \vdash\ [a/x]^* c \leftrightarrow [a/x]^* d : C} \qquad\qquad \frac{\Gamma\ \vdash\ a \leftrightarrow b : A \qquad \Delta,\ x : A\ \vdash\ c : C}{\Gamma,\ \Delta\ \vdash\ [a/x]^* c \leftrightarrow [b/x]^* c : C}$$

(There are no equations in Γ yet.)

Axiom 5.6 In this setting, we can say *how* urterms may be *formed*. Even though products of arbitrary pairs of objects may be formed in an equideductive category, for the time being we only introduce the $\times$ syntax for urtypes. Then we employ the symbols $\star : \mathbf{1}$, π_0, π_1 and $\langle\ ,\ \rangle$ in the usual way.

On the other hand, the λ-calculus is usually presented in a form that allows successive abstractions (λI), but we prefer to take all exponentials at Σ. A single λ may therefore bind any number of variables at the same time, whilst application may take multiple arguments:

$$\frac{\Gamma,\ \vec{x}:\vec{A}\ \vdash\ \sigma:\Sigma}{\Gamma\ \vdash\ \lambda\vec{x}{:}\vec{A}.\ \sigma:\Sigma^{\vec{A}}}\ \lambda I \qquad\qquad \frac{\Gamma\ \vdash\ \phi:\Sigma^{\vec{A}} \qquad \Delta\ \vdash\ \vec{a}:\vec{A}}{\Gamma,\ \Delta\ \vdash\ \phi\vec{a}:\Sigma}\ \lambda E$$

Using product urtypes, we may combine a list of arguments into a single one. Partial application (*i.e.* to a shorter list) may be achieved by padding it out with variables that are then re-abstracted.

Remark 5.7 Associated with each of the urtype connectives are
(a) the *introduction* and *elimination* rules for urterms, which we have just given;
(b) *beta*- and either *eta*- or *extensionality*-rules that specify the interchanges that make these operations inverse; and
(c) *interchange-transmitting* rules that say that they respect interchanges between urterms.

Axiom 5.8 The interchangeability rules for the singleton $\mathbf{1}$ and product $A\times B$ are

$$\frac{\Gamma\ \vdash\ a:\mathbf{1}}{\Gamma\ \vdash\ a\leftrightarrow\star:\mathbf{1}}\ \mathbf{1}\text{-ext} \qquad \frac{\Gamma\ \vdash\ p\leftrightarrow q:A\times B}{\Gamma\ \vdash\ \pi_0 p\leftrightarrow\pi_0 q:A}\ \times E_0{\leftrightarrow} \qquad \frac{\Gamma\ \vdash\ p\leftrightarrow q:A\times B}{\Gamma\ \vdash\ \pi_1 p\leftrightarrow\pi_1 q:B}\ \times E_1{\leftrightarrow}$$

$$\frac{\Gamma\ \vdash\ a:A \qquad \Gamma\ \vdash\ b:B}{\Gamma\ \vdash\ \pi_0\langle a,b\rangle\leftrightarrow a:A}\ \times\beta_0 \qquad\qquad \frac{\Gamma\ \vdash\ a:A \qquad \Gamma\ \vdash\ b:B}{\Gamma\ \vdash\ \pi_1\langle a,b\rangle\leftrightarrow b:B}\ \times\beta_1$$

$$\frac{\Gamma\ \vdash\ p,q:A\times B \qquad \Gamma\ \vdash\ \pi_0 p\leftrightarrow\pi_0 q:A \qquad \Gamma\ \vdash\ \pi_1 p\leftrightarrow\pi_1 q:B}{\Gamma\ \vdash\ p\leftrightarrow q:A\times B}\ \times\text{-ext}$$

Lemma 5.9 Interchangeability for $A\times B$ satisfies the interchange-transmitting and η-rules,

$$\frac{a_1\leftrightarrow a_2:A \qquad b_1\leftrightarrow b_2:B}{\langle a_1,b_1\rangle\leftrightarrow\langle a_2,b_2\rangle:A\times B}\ \times I{\leftrightarrow} \qquad\qquad \frac{p:A\times B}{\langle\pi_0 p,\pi_1 p\rangle\leftrightarrow p:A\times B}\ \times\eta$$

Proof In both cases, first use both $\times\beta_0$ and $\times\beta_1$, then transitivity of $\leftrightarrow$ and finally $\times$-ext. $\square$

Axiom 5.10 We need to be more careful about the rules for interchangeability of λ-terms *because we shall handle equality differently in equideductive logic*, where this *Axiom* will become *Lemma* 9.14.

$$\frac{\Gamma,\ \vec{x}:\vec{A}\ \vdash\ \sigma\leftrightarrow\tau:\Sigma}{\Gamma\ \vdash\ (\lambda\vec{x}{:}\vec{A}.\ \sigma)\leftrightarrow(\lambda\vec{x}{:}\vec{A}.\ \tau):\Sigma^{\vec{A}}}\ \lambda I{\leftrightarrow}$$

$$\frac{\Gamma\ \vdash\ \phi\leftrightarrow\psi:\Sigma^{\vec{A}} \quad \Delta\ \vdash\ \vec{a}:\vec{A}}{\Gamma,\ \Delta\ \vdash\ (\phi\vec{a}\,)\leftrightarrow(\psi\vec{a}\,):\Sigma}\ \lambda E{\leftrightarrow_0} \qquad \frac{\Gamma\ \vdash\ \phi:\Sigma^{\vec{A}} \quad \Delta\ \vdash\ \vec{a}\leftrightarrow\vec{b}:\vec{A}}{\Gamma,\ \Delta\ \vdash\ (\phi\vec{a}\,)\leftrightarrow(\phi\vec{b}\,):\Sigma}\ \lambda E{\leftrightarrow_1}$$

$$\frac{\Gamma\ \vdash\ \vec{a}:\vec{A} \qquad \Gamma,\ \vec{x}:\vec{A}\ \vdash\ \sigma:\Sigma}{\Gamma\ \vdash\ (\lambda\vec{x}{:}\vec{A}.\ \sigma)\vec{a}\leftrightarrow[\vec{a}/\vec{x}]^*\sigma:\Sigma}\ \lambda\beta \qquad\qquad \frac{\Gamma\ \vdash\ \phi:\Sigma^{\vec{A}}}{\Gamma\ \vdash\ (\lambda\vec{x}{:}\vec{A}.\ \phi\vec{x}\,)\leftrightarrow\phi:\Sigma^{\vec{A}}}\ \lambda\eta$$

6 Equivalence

Having described the restricted λ-calculus symbolically, we can consider its relationship to category theory. However, whilst the interpretation of the λ-calculus is usually given in a *cartesian closed* category, here we shall use the subcategory $\mathcal{A}$ of urspaces (Notation 4.9) in an *equideductive* one.

We first go from syntax to semantics. For a category $\mathcal{A}$ to be a suitable target for such an interpretation, it must of course have the relevant structure (products and powers of Σ), but it also has to have a *choice* of this amongst the many available isomorphic copies. We return to this rather distracting point at the end of the section.

Proposition 6.1 Let $\mathcal{A}$ be a category with chosen finite products and exponentials, together with a suitable interpretation of the base types and operation symbols of a particular language $\mathcal{L}$ as objects and morphisms of $\mathcal{A}$ that satisfy the particular interchanges. Then the restricted λ-calculus has an ***interpretation*** or ***denotation*** $[\![-]\!]$ in which interchangeable urterms in the calculus are denoted by the same morphism in the category.

Proof The urtypes and urcontexts are interpreted by structural recursion as follows:

(a) the interpretations of the base types of $\mathcal{L}$ as objects of $\mathcal{A}$ are given;

(b) the product and exponential urtypes in the symbolic calculus are interpreted by the structure of the same name in the category:

$$[\![\mathbf{1}]\!] \;\equiv\; \mathbf{1} \qquad [\![A \times B]\!] \;\equiv\; [\![A]\!] \times [\![B]\!] \qquad [\![\Sigma^{\vec{A}}]\!] \;\equiv\; \Sigma^{[\![A_1]\!]\times\cdots\times[\![A_n]\!]};$$

(c) each urcontext is interpreted as the product of the interpretations of its urtypes:

$$[\![\Gamma]\!] \;\equiv\; [\![x_1 : A_1, \ldots, x_n : A_n]\!] \;\equiv\; [\![A_1]\!] \times \cdots \times [\![A_n]\!].$$

Then an urterm in urcontext, $\Gamma \vdash a : A$, is interpreted as a morphism $[\![\Gamma]\!] \xrightarrow{[\![a]\!]} [\![A]\!]$ in $\mathcal{A}$, by structural recursion on its formation rules, as follows:

(d) if $x : A$ is one of the urtyped variables in the urcontext Γ then its interpretation is the relevant product projection, $[\![x]\!] : [\![\Gamma,\ x : A]\!] = [\![\Gamma]\!] \times [\![A]\!] \xrightarrow{\pi_1} [\![A]\!]$;

(e) weakening of the judgement $\Gamma \vdash a : A$ by an urtyped variable $y : B$ is obtained by pre-composition with the product projection, $[\![\Gamma, y : B]\!] = [\![\Gamma]\!] \times [\![B]\!] \xrightarrow{\pi_0} [\![\Gamma]\!] \xrightarrow{[\![a]\!]} [\![A]\!]$;

(f) contraction of a judgement $\Gamma, y : B, z : B \vdash a : A$ by identifying two variables of the same urtype is interpreted by pre-composing a diagonal,

$$[\![\Gamma, y : B]\!] = [\![\Gamma]\!] \times [\![B]\!] \xrightarrow{\Delta} [\![\Gamma]\!] \times [\![B]\!] \times [\![B]\!] = [\![\Gamma, y : B, z : B]\!] \xrightarrow{[\![a]\!]} [\![A]\!];$$

(g) exchange is given by switching the appropriate factors of the product;

(h) the interpretation of the cut rule in Definition 5.5(c) combines the maps

$$[\![\Gamma]\!] \xrightarrow{\quad[\![a]\!]\quad} [\![A]\!] \qquad \text{and} \qquad [\![\Delta]\!] \times [\![A]\!] \xrightarrow{\quad[\![b]\!]\quad} [\![B]\!]$$

into

$$[\![\Delta]\!] \times [\![\Gamma]\!] \xrightarrow{\quad[\![\Delta]\!] \times [\![a]\!]\quad} [\![\Delta]\!] \times [\![A]\!] \xrightarrow{\quad[\![b]\!]\quad} [\![B]\!]$$

using categorical product and composition;

(i) pairing and projections are interpreted using the categorical correspondence in Remark 5.1;

(j) the operation symbols of $\mathcal{L}$ have given interpretations as morphisms of the category, but note that we need to generate part of the interpretation in order to specify these "given interpretations";

(k) application (to variables) and λ-abstraction are interpreted by exponential transposition, as in Remark 5.1; and

(l) application to general expressions is derived from this using cut.

Finally, given a proof that two urterms are interchangeable in the calculus, we must show (again by induction on the structure of the proof) that they are denoted by the same morphism in the category. Recall that interchange judgements cannot (yet) depend on equational hypotheses.

(m) Reflexivity, symmetry and transitivity for morphisms are as for urterms;

(n) product satisfies $\times\beta_0$, $\times\beta_1$ and $\times$-extensionality because the categorical definition says that any pair $\langle a, b\rangle$ is the unique map for which $\pi_0\langle a, b\rangle = a$ and $\pi_1\langle a, b\rangle = b$;

(o) the interchange-transmitting rules $\times E_0\leftrightarrow$, $\times E_1\leftrightarrow$, $\lambda E\leftrightarrow_0$ and $\lambda E\leftrightarrow_1$ hold because $\times E_0$, $\times E_1$ and λE are defined by composition with π_0, π_1 and ev, but if pairs of maps are correspondingly the same in a category then so are their composites;

(p) the particular interchanges for the operation symbols of $\mathcal{L}$ are given to be valid in the category;

(q) the $\lambda\beta$-rule is expressed by the square on the left below: the clockwise composite is $\mathsf{ev}(\widetilde{[\![\sigma]\!]}, [\![a]\!])$ and the anticlockwise one is $[\![[a/x]^*\sigma]\!]$ by (h); this square commutes because the upper left triangle does by the properties of $\times$ and the lower one is Definition 4.3 of the transpose;

$$\begin{tikzcd}
\Gamma \arrow[r, "\langle\widetilde{[\![\sigma]\!]}, [\![a]\!]\rangle"] \arrow[d, "\langle \mathsf{id}, [\![a]\!]\rangle"'] & \Sigma^A \times A \arrow[d, "\mathsf{ev}"] \\
\Gamma \times A \arrow[ur, "\widetilde{[\![\sigma]\!]} \times A"] \arrow[r, "{[\![\sigma]\!]}"] & \Sigma
\end{tikzcd}
\qquad
\begin{tikzcd}
\Gamma \times A \arrow[r, "\widetilde{[\![\sigma]\!]} \times A"] \arrow[dr, "{[\![\sigma]\!]}"'] & \Sigma^A \times A \arrow[d, "\mathsf{ev}"] \\
& \Sigma
\end{tikzcd}$$

(r) the $\lambda\eta$-rule is the triangle on the right, which is also the definition of the transpose;

(s) for the structural rules applied to interchanges, in place of one of the arrows in 6.1(h), we have two that are the same, so they have the same composite with the other map. □

Remark 6.2 This interpretation may be developed without modification in the subcategory $\mathcal{A} \subset \mathcal{Q}$ of urspaces in an equideductive category, because $\mathcal{A}$ was required to be closed under the urtype-forming operations $\mathbf{1}$, $\times$ and $\Sigma^{(-)}$ that we need. When we extend the theory to equideductive logic in Section 11 it will be important that *all* objects $\Gamma \in \mathcal{Q}$ respect the universal properties of these type connectives. It will also be necessary to re-work the final section of the proof (parts m–s), because we shall handle equations there differently from interchangeability here. □

The category in which the language is interpreted need not come from outside, but may be obtained from the language itself.

Definition 6.3 The ***category of contexts and substitutions*** $\mathsf{Cn}^\lambda_\mathcal{L}$ has

(a) as *objects*, the urcontexts Γ;

(b) as *morphisms* $[\vec{b}/\vec{y}] : \Gamma \to \Delta \equiv [\vec{y} : \vec{B}]$, strings of interchangeability classes of urterms $\Gamma \vdash \vec{b} : \vec{B}$ or substitutions $[\vec{b}/\vec{x}]$;

(c) as *identity* on $\Gamma \equiv [\vec{x} : \vec{A}]$, the string $\vec{x}$ of variables;

(d) as *composite*, the substituted string $[\vec{b}/\vec{y}]^*\vec{c}$ of urterms or combined substitution $[\vec{c}/\vec{z}] \cdot [\vec{b}/\vec{y}] \equiv \left[[\vec{b}/\vec{y}]^*\vec{c}/\vec{z}\right]$.

Instead of defining its morphisms as strings $\vec{b}$ of urterms, [Tay99, §4.7] gives a more principled approach to the construction of $\mathsf{Cn}^\lambda_\mathcal{L}$. In this, the category is generated from an ***elementary sketch*** whose arrows are product projections and splittings of them, subject to an equivalence relation that is essentially the well known substitution lemma. This sketch is defined directly from the syntax of a language — the generating maps correspond to the urtypes and urterms — even when this has dependent types [Tay99, Ch. VIII].

Lemma 6.4 The structure $\mathsf{Cn}^\lambda_\mathcal{L}$ is a category that has a choice of products and exponentials.

Proof Axiom 5.5 provides the categorical structure: interchangeability is an equivalence relation that is respected by cut, which is itself associative and behaves appropriately for variables. The products of single-variable urcontexts $[x : A] \times [y : B]$ are given by products of the urtypes, $[p : A \times B]$. Alternatively, products of multiple-variable urcontexts $[\vec{x} : \vec{A}] \times [\vec{y} : \vec{B}]$ are obtained by (renaming any common variables and) concatenating the urcontexts, $[\vec{x} : \vec{A}, \vec{y} : \vec{B}]$. The exponential $\Sigma^{[\vec{x}:\vec{A}]}$ is $[\phi : \Sigma^{\Pi\vec{A}}]$. $\square$

Theorem 6.5 $\mathsf{Cn}^\lambda_\mathcal{L}$ is the ***classifying category*** for the restricted λ-calculus generated by the language $\mathcal{L}$:

(a) it is itself a category with chosen products and $\Sigma^{(-)}$;

(b) it has an interpretation of the calculus;

(c) any interpretation of the calculus in a category $\mathcal{A}$ with chosen $\times$ and $\Sigma^{(-)}$ extends to a functor $\mathsf{Cn}^\forall_\mathcal{L} \to \mathcal{Q}$ that preserve this structure, uniquely up to unique natural isomorphism; and

(d) any such structure-preserving functor restricts to an interpretation of the language in $\mathcal{Q}$.

Proof Every object of $\mathsf{Cn}^\lambda_\mathcal{L}$ (context) is a finite product of single-variable urcontexts $[x : A]$, each of which corresponds to an urtype, whilst all of the functors in question preserve products. It is therefore enough to consider objects or urcontexts like $[x : A]$ and morphisms into them, which are single urterms $\Gamma \vdash a : A$. The extension of the interpretation from single urtypes and urterms to urcontexts depends on the choice of products, but any two such choices are uniquely naturally isomorphic. $\square$

For the inverse construction, we need to augment the λ-calculus with a *name* for each object, morphism, product and exponential.

Definition 6.6 Let $\mathcal{A}$ be any category with products and exponentials (for example the subcategory $\mathcal{A} \subset \mathcal{Q}$ of urspaces in an equideductive category). Then the ***proper language*** of $\mathcal{A}$ has

(a) a *base type* $\ulcorner A \urcorner$ for each object $A \in \mathcal{A}$;

(b) an *operation symbol* $x : \ulcorner A \urcorner \vdash \ulcorner f \urcorner x : \ulcorner B \urcorner$ for each morphism $f : A \to B$ in $\mathcal{A}$;

(c) a *particular interchange* $x : \ulcorner A \urcorner \quad \vdash \quad \ulcorner g \urcorner(\ulcorner f \urcorner x) \;\leftrightarrow\; \ulcorner g \cdot f \urcorner x : \ulcorner C \urcorner$ for each composite $A \xrightarrow{f} B \xrightarrow{g} C$ in $\mathcal{A}$;

(d) an *operation symbol* $x : \ulcorner A \urcorner,\ y : \ulcorner B \urcorner \;\vdash\; \mathsf{pair}(x, y) : \ulcorner A \times B \urcorner$ with

(e) *particular interchanges*

$$\begin{array}{lcl}
x : \ulcorner A \urcorner,\ y : \ulcorner B \urcorner & \vdash & \ulcorner \pi_0 \urcorner\big(\mathsf{pair}(x,y)\big) \;\leftrightarrow\; x : \ulcorner A \urcorner \\
x : \ulcorner A \urcorner,\ y : \ulcorner B \urcorner & \vdash & \ulcorner \pi_1 \urcorner\big(\mathsf{pair}(x,y)\big) \;\leftrightarrow\; y : \ulcorner B \urcorner \\
p : \ulcorner A \times B \urcorner & \vdash & \mathsf{pair}\left(\ulcorner \pi_0 \urcorner p, \ulcorner \pi_1 \urcorner p\right)\big) \;\leftrightarrow\; p : \ulcorner A \times B \urcorner
\end{array}$$

for each pair of objects $A, B \in \mathcal{A}$; and

(f) an *operation symbol* $\phi : \Sigma^{\ulcorner A \urcorner} \;\vdash\; \mathsf{abs}\,\phi : \ulcorner \Sigma^{A} \urcorner$ with

(g) *particular interchanges*

$$\begin{array}{lll} \phi : \Sigma^{\ulcorner A \urcorner},\; x : \ulcorner A \urcorner & \vdash & \ulcorner \mathsf{ev} \urcorner\big(\mathsf{abs}\,\phi, x\big) \;\leftrightarrow\; \phi x : \Sigma \\ f : \ulcorner \Sigma^{A} \urcorner & \vdash & \mathsf{abs}\,\big(\lambda x{:}\ulcorner A \urcorner.\, \ulcorner \mathsf{ev} \urcorner(f, x)\big) \;\leftrightarrow\; f : \ulcorner \Sigma^{A} \urcorner \end{array}$$

for each object $A \in \mathcal{A}$.

Remark 6.7 Many authors call this the *internal* language, but this name is not consistent with other categorical terminology. One can formalise notions of "language" mathematically just as one can a group. Such formalisations admit interpretations in categories with suitable structure, for example there are internal groups in any category with products, the leading example being *topological* groups. Likewise there are internal models of mathematically formalised notions of language in appropriate categories, which would in particular also have to have free monoid functors. The resulting notion of *internal language* could, for example, be useful in a categorical study of Gödel's theorem. However, this is not what we are using here.

Theorem 6.8 For any category $\mathcal{A}$ with finite products and $\Sigma^{(-)}$, the functor $\ulcorner - \urcorner : \mathcal{A} \to \mathsf{Cn}^{\lambda}_{\mathcal{L}}$ defines a weak equivalence with the category of contexts and substitutions of its proper language. That is, $\ulcorner - \urcorner$ is full, faithful and essentially surjective. It is a strong equivalence, having a pseudo-inverse $[\![-]\!] : \mathsf{Cn}^{\lambda}_{\mathcal{L}} \to \mathcal{A}$ with

$$\epsilon_A : [\![x : \ulcorner A \urcorner]\!] \;=\; A \qquad \text{and} \qquad \eta_\Gamma : \ulcorner [\![\Gamma]\!] \urcorner \;\cong\; \Gamma,$$

iff $\mathcal{A}$ has a *choice* of the necessary structure.

Proof This is discussed in detail in [Tay99, §7.6]. so we just sketch the strategy here.

Suppose first that $\mathcal{A}$ has a choice of structure. Then requirements like $[\![\ulcorner A \urcorner]\!] = A$ provide the base cases of the recursive definition of $[\![-]\!] : \mathsf{Cn}^{\lambda}_{\mathcal{L}} \to \mathcal{A}$ in Proposition 6.1. The isomorphism $\ulcorner [\![\Gamma]\!] \urcorner \cong \Gamma$ is also defined from the particular operation-symbols and interchanges by recursion on (the proof that the urtypes are well formed in) Γ.

Naturality of this isomorphism with respect to Γ is not trivial: it is equivalent to $\ulcorner - \urcorner : \mathcal{A} \to \mathsf{Cn}^{\lambda}_{\mathcal{L}}$ being full and faithful and needs to be proved for each connective. In our case ($\times$ and $\Sigma^{(-)}$) this can be done using the normal form.

Without the *choice* of structure, we must show that any $x : \ulcorner A \urcorner \vdash fx : \ulcorner B \urcorner$ is $fx = \ulcorner g \urcorner x$ for some unique morphism $g : A \to B$ in $\mathcal{A}$, and each urcontext Γ has $\Gamma \cong \ulcorner A \urcorner$ for some object $A \in \mathcal{A}$. The proof for a particular urterm or urcontext is a *finite part* of the general result, *i.e.* it requires the existence of *finitely many* instances of the categorical structure in $\mathcal{A}$. However, since the identities of these instances are not exported from the proof in the statement of the theorem, no *choice* of them is needed. $\square$

Corollary 6.9 The interpretation $[\![-]\!] : \mathsf{Cn}^{\lambda}_{\mathcal{L}} \to \mathcal{A}$ is full and faithful: any map $f : [\![\Gamma]\!] \to [\![A]\!]$ in $\mathcal{A}$ is the interpretation of an urterm

$$\Gamma \equiv [\vec{s} : \vec{C}] \quad \vdash \quad a \;\equiv\; \eta_A^{-1}\left(\ulcorner e \urcorner (\eta_\Gamma\, \vec{z})\right) : A.$$

$$\begin{tikzcd} {[\![\Gamma]\!]} \arrow[d, "f"'] & \Gamma \arrow[r, "\eta_\Gamma", "\cong"'] \arrow[d, dotted, "a"'] & \ulcorner [\![\Gamma]\!] \urcorner \arrow[d, "\ulcorner f \urcorner"] \\ {[\![A]\!]} & A \arrow[r, "\eta_A", "\cong"'] & \ulcorner [\![A]\!] \urcorner \end{tikzcd}$$

If $\Gamma \vdash b : A$ has the same interpretation then the interchange $\Gamma \vdash a \leftrightarrow b : A$ is provable from the proper language. $\square$

7 Sobriety

The contravariant functor $\Sigma^{(-)}$ warrants deeper study. It is self-adjoint on the full subcategory of exponentiable objects on which it is defined (Proposition 4.14), so the covariant double exponential $\Sigma^{\Sigma^{(-)}}$ is part of a monad, whose unit η_X is $x \mapsto \lambda\phi.\ \phi x$ in λ-notation (Definition 4.3). The situation where η_A is an equaliser was investigated in [A], introducing a new term-forming operation called focus into the symbolic language. This built on the notion of *repleteness* in synthetic domain theory that was introduced by Martin Hyland [Hyl91] and relied on an orthogonality property similar to our Proposition 2.9.

Our approach in this paper is different from earlier one in [A]. Whereas that constructed a *new* category to make the objects sober, in this section we shall work within a *given* equideductive category and prove *as a theorem* that every exponentiable object of it is sober.

The symbolic language that we have so far (the restricted λ-calculus) is not strong enough to carry the argument. In particular, it cannot express equalisers, so we still have to work in a categorical style here. We can use the proper restricted λ-calculus of the subcategory $\mathcal{A} \subset \mathcal{Q}$ of urspaces for those parts of the development that only use exponentiable objects.

Remark 7.1 Recall from [A] that if the exponentials Σ^A and Σ^B exist then they are algebras for the $\Sigma^{\Sigma^{(-)}}$ monad. Moreover, for any $f : A \to B$, the map $H \equiv \Sigma^f$ is a ***homomorphism***. This means that it makes the square on the left commute, by naturality of Σ^η with respect to f:

$$\begin{tikzcd} \Sigma^3 B \equiv \Sigma^{\Sigma^{\Sigma^B}} \arrow[r, "\Sigma^{\Sigma^H}"] \arrow[d, "\Sigma^{\eta_B}"'] & \Sigma^3 A \equiv \Sigma^{\Sigma^{\Sigma^A}} \arrow[d, "\Sigma^{\eta_A}"] \\ \Sigma^B \arrow[r, "H"] & \Sigma^A \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "P"] & \Sigma^{\Sigma^B} \arrow[r, shift left, "\eta\Sigma^2 B"] \arrow[r, shift right, "\Sigma^2\eta B"'] & \Sigma^4 B \equiv \Sigma^{\Sigma^{\Sigma^{\Sigma^B}}} \end{tikzcd}$$

Equivalently, the transpose $P \equiv \tilde{H}$ has the same composite with $\eta\Sigma^2 B$ as with $\Sigma^2\eta B$. The easiest way to see this is to use the restricted λ-calculus from the previous section, because then H and P differ only in the order of their arguments. Since the towers of exponentials are rather unwieldy, we abbreviate them as Σ^n.

We can turn the requirement that the composites be the same into a symbolic interchange:

Definition 7.2 A term $\Gamma \vdash P : \Sigma^{\Sigma^A}$ of the restricted λ-calculus is called ***prime*** for the exponentiable object A if it is provable that

$$\Gamma,\ \Phi : \Sigma^3 A \quad \vdash \quad \Phi P \ \leftrightarrow\ P\big(\lambda x.\ \Phi(\lambda\phi.\ \phi x)\big) : \Sigma.$$

In particular, any term $\Gamma \vdash a : A$ of exponentiable type gives rise the prime $P \equiv \lambda\phi.\ \phi a$.

Lemma 7.3 For $\Delta, x : A \vdash fx : B$, if $\Gamma \vdash P : \Sigma^{\Sigma^A}$ is prime then so too is

$$\Gamma,\ \Delta \quad \vdash \quad Q \ \equiv\ (\Sigma^{\Sigma^f})P \ \equiv\ \lambda\psi.\ P\big(\lambda x.\ \psi(fx)\big) : \Sigma^{\Sigma^B}.$$

Proof Categorically, this is because $\eta\Sigma^2(-)$ and $\Sigma^2\eta_{(-)}$ are natural transformations, *i.e.* the two squares on the right commute:

$$\begin{tikzcd}
 & \Delta\times A \arrow[r, "\eta_A\cdot\pi_1"] \arrow[dd, "f"] & \Sigma^{\Sigma^A} \arrow[r, shift left, "\eta\Sigma^2 A"] \arrow[r, shift right, "\Sigma^2\eta_A"'] \arrow[dd, "\Sigma^{\Sigma^f}"'] & \Sigma^4 A \arrow[dd, "\Sigma^4 f"] \\
\Gamma \arrow[urr, "P"] \arrow[drr, "Q"'] & & & \\
 & B \arrow[r, "\eta_B"'] & \Sigma^{\Sigma^B} \arrow[r, shift left, "\eta\Sigma^2 B"] \arrow[r, shift right, "\Sigma^2\eta_B"'] & \Sigma^4 B
\end{tikzcd}$$

Symbolically, consider the urcontext $\Gamma, \Delta, \Psi : \Sigma^3 B$. Then

$$\begin{array}{rcl}
\Psi\big((\Sigma^{\Sigma^f})P\big) & \equiv & \Psi\big(\lambda\psi.\ P(\lambda x.\ \psi b)\big) \\
 & \equiv & \Phi P \ \leftrightarrow\ P\big(\lambda x.\ \Phi(\lambda\phi.\ \phi x)\big) \\
 & \equiv & P\big(\lambda x.\ \Psi\big(\lambda\psi.\ (\lambda\phi.\ \phi x)(\lambda x.\ \psi b)\big)\big) \\
 & \leftrightarrow & P\big(\lambda x.\ \Psi(\lambda\psi.\ \psi b)\big) \ \equiv\ \big((\Sigma^{\Sigma^f})P\big)\big(\lambda y.\ \Psi.(\lambda\psi.\ \psi y)\big),
\end{array}$$

using primality of P with respect to the expression Φ that is defined by the second line. $\square$

Definition 7.4 For any exponentiable object A of an equideductive category, let $\bar{A} \in \mathcal{Q}$ be the *subspace* of primes for A, which is defined by the equaliser

$$\begin{tikzcd}
A \arrow[d, "\epsilon"'] \arrow[dr, "\eta_A"] & & \\
\bar{A} \arrow[r, tail, "j"] & \Sigma^2 A \arrow[r, shift left, "\eta\Sigma^2 A"] \arrow[r, shift right, "\Sigma^2\eta A"'] & \Sigma^4 A
\end{tikzcd}$$

Then $j : \bar{A} \rightarrowtail \Sigma^2 A$ is in $\mathcal{M}$ by Lemma 4.7, and we say that A is ***sober*** if the mediator $\epsilon : A \to \bar{A}$ is an isomorphism.

The following key lemma belongs to the tradition of synthetic domain theory and may be needed in some other setting in future. We therefore note that it holds *whenever the Definition is meaningful and Σ is injective with respect to $j \times Y$.*

Lemma 7.5 The exponential $\Sigma^{\bar{A}}$ exists in the presheaf topos $\mathbf{Set}^{\mathcal{Q}^{\mathrm{op}}}$ and $\Sigma^\epsilon : \Sigma^{\bar{A}} \cong \Sigma^A$. Hence $\epsilon : A \to \bar{A}$ is Σ-epi in $\mathcal{Q}$.

Proof The Yoneda embedding preserves the exponentials $\Sigma^n A$ and the equaliser in the Definition and all exponentials exist in a topos. Then

$$\begin{tikzcd}
\Sigma^A \arrow[r, dotted, shift left, "\eta\Sigma^A"] \arrow[d, dotted, shift right] & \Sigma^3 A \arrow[l, two heads, shift left, "\Sigma\eta_A"] \arrow[r, dotted, shift left=2, "\eta\Sigma^3 A"] \arrow[dl, "\Sigma^j"'] & \Sigma^5 A \arrow[l, "\Sigma\eta\Sigma^2 A"] \arrow[l, shift left=2, "\Sigma^3\eta A"] \\
\Sigma^{\bar{A}} Y \arrow[u, shift right, "\Sigma^\epsilon"] & Y \arrow[l, shift right, "\tilde{\phi}"'] \arrow[l, shift left, "\tilde{\psi}"] \arrow[u, shift left, "\tilde{\Phi}"] \arrow[u, shift right, "\tilde{\Psi}"'] &
\end{tikzcd}$$

the top row is a split coequaliser. Therefore there is a mediator $k : \Sigma^A \hookrightarrow \Sigma^{\bar{A}}$ such that

$$\Sigma^j \;=\; k \cdot \Sigma^{\eta_A}, \qquad \Sigma^{\eta_A} \;=\; \Sigma^e \cdot \Sigma^j \qquad \text{and} \qquad \Sigma^e \cdot k \;=\; \mathsf{id}_{\Sigma^A}.$$

Also, injectivity of Σ means that the natural transformation

$$\Sigma^j(Y) : \Sigma^3 A(\Gamma) \;\equiv\; \mathcal{Q}\big(\Gamma \times \Sigma^{\Sigma^A}, \Sigma\big) \longrightarrow \Sigma^{\bar{A}}(\Gamma) \;\equiv\; \mathcal{Q}(\Gamma \times \bar{A}, \Sigma)$$

is componentwise surjective, *cf.* Lemma 2.12. Hence $k \cdot \Sigma^\epsilon$ is componentwise the same as the identity, so it is the identity. In particular, the natural transformation

$$\Sigma^\epsilon(Y) : \Sigma^{\bar{A}}(Y) \;\equiv\; \mathcal{Q}(\bar{A} \times Y) \longrightarrow \Sigma^A(Y) \;\equiv\; \mathcal{Q}(A \times Y)$$

given by composition with $\epsilon \times Y$ is mono, which means that ϵ is Σ-epi by Lemma 2.13. $\square$

We can also give a bare-hands proof that avoids the Yoneda embedding and set theory. We need to be careful because Y and *â priori* $\bar{A}$ need not themselves be exponentiable, but they do have to respect the iterated exponentials and equaliser that we are using.

Lemma 7.6 The map $\epsilon : A \to \bar{A}$ is Σ-epi.

$$\begin{tikzcd}
A \times Y \arrow[rr, "\eta_A \times Y"] \arrow[d, "\epsilon \times Y"'] & & \Sigma^{\Sigma^A} \times Y \arrow[r, shift left, "\eta\Sigma^2 A \times Y"] \arrow[r, shift right, "\Sigma^2 \eta A \times Y"'] \arrow[d, shift right, "\Phi"'] \arrow[d, shift left, "\Psi"] & \Sigma^4 A \times Y \\
\bar{A} \times Y \arrow[urr, "j \times Y"] \arrow[rr, shift left, "\phi"] \arrow[rr, shift right, "\psi"'] & & \Sigma &
\end{tikzcd}$$

Proof For Definition 2.8 we must show that, for any object $Y \in \mathcal{Q}$ and morphisms $\phi, \psi : \bar{A} \times Y \rightrightarrows \Sigma$, if the composites $\phi \cdot (\epsilon \times Y)$ and $\psi \cdot (\epsilon \times Y)$ are the same (θ) then so already were ϕ and ψ.

By Lemma 4.7, the equaliser inclusion $j \times Y$ is in $\mathcal{M}$, so by injectivity of Σ (Axiom 4.2) the maps ϕ, ψ lift to Φ, Ψ, so that $\Phi \cdot (j \times Y)$ and $\Psi \cdot (j \times Y)$ are the same as ϕ and ψ respectively. The upper triangle above commutes by construction, so all paths from $A \times Y$ to Σ have the same composite, θ.

Consider the exponential transpose $\tilde{\Phi}$ of Φ (Definition 4.3), which is defined by the lower right commutative triangle below:

$$\begin{tikzcd}
A \times \Sigma^3 A \arrow[dd, "A \times \Sigma^{\eta_A}"'] \arrow[dr, "\eta_A \times \Sigma^3 A"] & & A \times Y \arrow[ll, "A \times \tilde{\Phi}"'] \arrow[dd, "\theta"] \arrow[dr, "\eta_A \times Y"] & \\
& \Sigma^{\Sigma^A} \times \Sigma^3 A \arrow[dr, "\mathsf{ev}'"] & & \Sigma^{\Sigma^A} \times Y \arrow[ll, "\Sigma^{\Sigma^A} \times \tilde{\Phi}"'] \arrow[dl, "\Phi"] \\
A \times \Sigma^A \arrow[rr, "\mathsf{ev}'"] & & \Sigma &
\end{tikzcd}$$

The triangle on the right commutes by definition of θ. The parallelogram that overlaps it commutes because it is the product of the morphisms η_A and $\tilde{\Phi}$. The big triangle on the lower left commutes by a λ-calculation that is valid because all of its vertices are exponentiable and the common composite takes (a, Ξ) to $\Xi(\lambda\xi.\, \xi a)$. Hence the square commutes from $A \times Y$ to Σ, which makes

$\Sigma^{\eta_A} \cdot \tilde{\Phi}$ the transpose of θ. The same is true of $\Sigma^{\eta_A} \cdot \tilde{\Psi}$, but transposes are unique, so the two triangles on the left below commute:

$$\begin{tikzcd}
\bar{A} \times Y \arrow[r, shift left, "\bar{A} \times \tilde{\Phi}"] \arrow[r, shift right, "\bar{A} \times \tilde{\Psi}"'] \arrow[dr, "\bar{A} \times \tilde{\theta}"'] & \bar{A} \times \Sigma^3 A \arrow[r, "j \times \Sigma^3 A"] \arrow[d, "\bar{A} \times \Sigma^{\eta_A}"] & \Sigma^{\Sigma^A} \times \Sigma^3 A \arrow[r, shift left, "\eta \Sigma^{\Sigma^A} \times \Sigma^3 A"] \arrow[r, shift right, "\Sigma^2 \eta A \times \Sigma^3 A"'] \arrow[d, "\Sigma^{\Sigma^A} \times \Sigma^{\eta_A}"'] \arrow[dr, "\mathsf{ev}'"] & \Sigma^4 A \times \Sigma^3 A \arrow[d, "\mathsf{ev}"] \\
& \bar{A} \times \Sigma^A \arrow[r, "j \times \Sigma^A"] & \Sigma^{\Sigma^A} \times \Sigma^A \arrow[r, "\mathsf{ev}"] & \Sigma
\end{tikzcd}$$

The square in the middle commutes because it is the product of the morphisms j and Σ^{η_A}. On the right, the lower square (*via* $\Sigma^2 \eta A \times \Sigma^3 A$) and the upper triangle (involving $\eta \Sigma^2 A \times \Sigma^3 A$ and ev') commute by λ-calculations that are valid because all the vertices are exponentiable. The common composite of the former takes (F, Ξ) to $F\big(\lambda a.\, \Xi(\lambda \xi.\, \xi a)\big)$ whilst the latter defines η (Definition 4.3).

By construction, j has the same composite with $\eta \Sigma^2 A$ as with $\Sigma^2 \eta A$, so the composites from $\bar{A} \times \Sigma^3 A$ to $\Sigma^4 A \times \Sigma^3 A$ along the top are the same. Hence all paths from $\bar{A} \times Y$ to Σ have the same composite. We rewrite this along the lower path around the diagram below:

$$\begin{tikzcd}
\bar{A} \times Y \arrow[r, "j \times A"] \arrow[d, "\bar{A} \times \tilde{\Phi}"'] & \Sigma^{\Sigma^A} \times Y \arrow[r, "\Phi"] \arrow[d, "\Sigma^{\Sigma^A} \times \tilde{\Phi}"'] & \Sigma \\
\bar{A} \times \Sigma^3 A \arrow[r, "j \times \Sigma^{\Sigma^A}"] & \Sigma^{\Sigma^A} \times \Sigma^3 A \arrow[ur, "\mathsf{ev}'"'] &
\end{tikzcd}$$

in which the triangle is the definition of $\tilde{\Phi}$ and the square is the product of this and j. The composite along the top is ϕ by construction, but this is the same as ψ since we could have used Ψ instead of Φ. $\square$

Proposition 7.7 Every exponentiable object of an equideductive category is sober.

$$\begin{tikzcd}
A \arrow[rrrr, two heads, "\epsilon"] \arrow[dd, "f \equiv \mathsf{id}"'] & & & & \bar{A} \arrow[dlll, "j"'] \arrow[ddllll, dotted] \arrow[dd, "g"] \\
& \Sigma^{\Sigma^A} \arrow[rr, "\Sigma^{\Sigma^i}"'] & & \Sigma^3 B \arrow[dr, "\Sigma^{\eta_B}"] & \\
A \arrow[ur, "\eta_A"] \arrow[rr, tail, "i"] & & \Sigma^B \arrow[ur, "\eta_{\Sigma^B}"] \arrow[rr, "\mathsf{id}"] & & \Sigma^B
\end{tikzcd}$$

Proof We must find the inverse of $\epsilon : A \to \bar{A}$. Since there are enough injectives (Axiom 4.6), there is a partial product inclusion ($\mathcal{M}$-map) $i : A \rightarrowtail \Sigma^B$. Then ϵ and i form a square with $f \equiv \mathsf{id}$ and $g \equiv \Sigma^{\eta_B} \cdot \Sigma^{\Sigma^i} \cdot j$ that commutes by naturality of η and its unit equation. By the previous lemma, ϵ is Σ-epi, so by orthogonality (Proposition 2.9) there is a unique fill-in $\bar{A} \to A$. This is the inverse of ϵ since we already know that it is Σ-epi. $\square$

Corollary 7.8 The map $\eta : A \to \Sigma^{\Sigma^A}$ is a regular mono, so it is in $\mathcal{M}$. $\square$

Lemma 7.9 A map $f : A \to B$ between exponentiable objects in an equideductive category is epi iff $\Sigma^f : \Sigma^B \to \Sigma^A$ is mono and this case $f \times C$ is also epi.

Proof The target X of the pair has an $\mathcal{M}$-map (mono) into an urtype Σ^C by Axiom 4.6. Then the double transpose of the diagram on the left gives the result:

$$\begin{tikzcd} A \arrow[r, "f"] & B \arrow[r, shift left, "h"] \arrow[r, shift right, "k"'] & X \arrow[r, tail] & \Sigma^C \end{tikzcd} \qquad \begin{tikzcd} \Sigma^A & \Sigma^B \arrow[l, "\Sigma^f"'] & C \arrow[l, shift right, "\tilde{h}"'] \arrow[l, shift left, "\tilde{k}"] \end{tikzcd}$$

It follows easily that $f \times C$ is also epi. $\square$

Theorem 7.10 The functor $\Sigma^{(-)} : \mathcal{B} \to \mathcal{B}^{\mathsf{op}}$ ***reflects invertibility***: if Σ^f is invertible, so is f.

$$\begin{tikzcd} \Sigma^B \arrow[d, shift left, "\Sigma^f"] \\ \Sigma^A \arrow[u, shift left, "G"] \arrow[u, phantom, "\cong"] \end{tikzcd} \qquad \begin{tikzcd} B \arrow[d, shift left, dotted] \arrow[drr, "\tilde{G}"] & & & \\ A \arrow[u, shift left, "f"] \arrow[rr, tail, "\tilde{\mathsf{id}} \equiv \eta_A"] & & \Sigma^{\Sigma^A} \arrow[r, shift left] \arrow[r, shift right] & \Sigma^4 A \end{tikzcd}$$

Proof Let $f : A \to B$ with $G : \Sigma^A \to \Sigma^B$ inverse to Σ^f. The double exponential transpose of $\Sigma^f \cdot G = \mathsf{id}$ is $\tilde{G} \cdot f = \tilde{\mathsf{id}} = \eta_A$, which has the same composite with $\Sigma^2 \eta_A$ and $\eta \Sigma^{\Sigma^A}$. Since f is epi by the Lemma, $\tilde{G}$ also has the same composite, so it factors through the equaliser, providing f^{-1}. $\square$

8 The sober lambda calculus

We can build the categorical notion of sobriety into the syntactic calculus by adding a new term-forming operation. The premises of the formation rules are the term P itself and the interchange judgement that says that this is prime. This extension potentially makes the proof theory of the calculus much more complicated, so we devote a lot of effort to eliminating these difficulties. In doing this, it will be useful to distinguish these more general *terms* from the *urterms* of the restricted λ-calculus (Section 5), in which focus was not allowed.

Axiom 8.1 The ***sober λ-calculus*** adds a new term-forming operation focus to Axiom 5.2(c) of the restricted λ-calculus. For any *base* type A, its introduction rule is

$$\frac{\Gamma \ \vdash \ P : \Sigma^{\Sigma^A} \qquad \Gamma,\ \Phi : \Sigma^3 A \ \vdash \ \Phi P \ \leftrightarrow \ P\big(\lambda x.\, \Phi(\lambda \phi.\, \phi x)\big)}{\Gamma \ \vdash \ \mathsf{focus}_A\, P : A}$$

so P must be prime. The β-rule, also for prime P, is

$$\Gamma,\ \phi : \Sigma^A \ \vdash \ \phi(\mathsf{focus}_A\, P) \ \leftrightarrow \ P\phi : \Sigma$$

and the η-rule is $\qquad x : A \ \vdash \ \mathsf{focus}_A(\lambda \phi.\, \phi x) \ \leftrightarrow \ x : A.$

Instead of asserting interchange-transmission rules directly, we use

$$\frac{\Gamma \ \vdash \ a : A \qquad \Gamma \ \vdash \ b : A \qquad \Gamma \ \vdash \ \lambda \phi.\, \phi a \ \leftrightarrow \ \lambda \phi.\, \phi b}{\Gamma \ \vdash \ a \leftrightarrow b} \qquad \mathrm{T}_0$$

We also extend

(a) Axioms 5.6 and 5.8 for pairing ($\times I$) and

(b) Axioms 5.4(a–d) and 5.5(a,b), the structural rules for urterms and interchanges *with the exception of cut* to terms involving focus. Hence we may form $\langle \mathsf{focus}\, P, \ldots\rangle$ but not apply π_0, π_1, λ or ev to focus.

Remark 8.2 Notice that there is a restriction on the applicability of the *introduction* rule for focus, just as there was on λI (namely that λ may only be applied to urterms of type Σ) in Axiom 5.6. However, once the term $\mathsf{focus}\, P$ has been validly formed, it (or, rather, its β-rule) may be *used* without further restriction.

Example 8.3 In Theorem 7.10, $f^{-1}b = \mathsf{focus}\,(\lambda\phi.\ G\phi b)$.

Lemma 8.4 The focus operation transmits interchanges ($\mathsf{focus}\leftrightarrow$) and also satisfies

$$\frac{a \ \leftrightarrow\ \mathsf{focus}\, P}{\overline{(\lambda\phi.\ \phi a)\ \leftrightarrow\ P}} \qquad \text{and} \qquad \frac{\mathsf{focus}\, P\ \leftrightarrow\ \mathsf{focus}\, Q}{\overline{P\ \leftrightarrow\ Q}}$$

Proof If $\Gamma \vdash P \leftrightarrow Q : \Sigma^{\Sigma^A}$ then

$$\Gamma \ \vdash\ \lambda\phi.\ \phi(\mathsf{focus}\, P)\ \leftrightarrow\ \lambda\phi.\ P\phi\ \leftrightarrow\ P\ \leftrightarrow\ Q\ \leftrightarrow\ \lambda\phi.\ Q\phi\ \leftrightarrow\ \lambda\phi.\ \phi(\mathsf{focus}\, Q),$$

so the T_0-rule applies and we have $\Gamma \vdash \mathsf{focus}\, P \leftrightarrow \mathsf{focus}\, Q$. The other parts follow from the β and η-rules. □

Remark 8.5 Instead of axiomatising them directly, we treat the other features of the calculus as definitional extensions:
(a) application of λ-terms and operation symbols to focus-terms,
(b) focus at product types and therefore $\pi_i(\mathsf{focus}\, R)$,
(c) focus at exponential types and therefore $\lambda x.\ (\mathsf{focus}\, \mathcal{P})$,
(d) cut or substitution of terms for variables and
(e) cut for interchanges.

Remark 8.6 We use $\mathsf{focus}\,\beta$ to define λ-application to focus-terms:

$$\phi(\mathsf{focus}\, P)\ \equiv\ P\phi$$

$$(\lambda x.\ \sigma)(\mathsf{focus}\, P)\ \equiv\ P(\lambda x.\ \sigma)\ \equiv\ [\mathsf{focus}\, P/x]^*\sigma.$$

Similarly, application of an operation-symbol is given by

$$r(\mathsf{focus}\, P)\ \equiv\ \mathsf{focus}\,\big(\lambda\psi.\ P(\lambda x.\ \psi(rx))\big).$$

These symbols and λ-terms may have more than one argument and we have allowed pairing of terms that involve focus. However, since focus itself has only been defined for base types, we must extract the arguments one at a time, for example,

$$\theta\langle \mathsf{focus}\, P, \mathsf{focus}\, Q\rangle\ \equiv\ P\big(\lambda x.\ \theta\langle x, \mathsf{focus}\, Q\rangle\big)\ \equiv\ \begin{cases} P\big(\lambda x.\ Q(\lambda y.\ \theta\langle x, y\rangle)\big) \\ Q\big(\lambda y.\ P(\lambda x.\ \theta\langle x, y\rangle)\big), \end{cases}$$

where we have a choice about how much of the surrounding expression is regarded as being applied $\mathsf{focus}\, Q$, but they are interchangeable because P and Q are prime. This is an aspect of the subtle interaction of the $\Sigma^{\Sigma^{(-)}}$ with products that is explored at length in [A] and [Füh99, Sel01, Thi97].

Lemma 8.7 $\mathsf{focus}_{\mathbf{1}}\, P \equiv \star$ and

$$\mathsf{focus}_{A\times B}\, R\ \equiv\ \big\langle\, \mathsf{focus}_A(\lambda\phi.\ R\big(\lambda p.\ \phi(\pi_0 p)\big)), \mathsf{focus}_B(\lambda\psi.\ R\big(\lambda p.\ \psi(\pi_1 p)\big))\,\big\rangle.$$

Proof Let $R : \Sigma^2(A \times B)$ be prime. Then

$$P \;\equiv\; (\Sigma^2\pi_0)R \;\equiv\; \lambda\phi.\, R(\lambda xy.\, \phi x) : \Sigma^{\Sigma^A} \quad \text{and} \quad Q \;\equiv\; (\Sigma^2\pi_1)R \;\equiv\; \lambda\psi.\, R(\lambda xy.\, \psi y) : \Sigma^{\Sigma^B}$$

are also prime by Lemma 7.3. To show that this satisfies the extended $\mathsf{focus}\,\beta$-rule, let $\theta : \Sigma^{A\times B}$; then focus-elimination for P and then Q as above gives

$$\theta\langle \mathsf{focus}\, P, \mathsf{focus}\, Q\rangle \;\leftrightarrow\; P\big(\lambda x.\, \theta\langle x, \mathsf{focus}\, Q\rangle\big) \;\leftrightarrow\; P\big(\lambda x.\, Q(\lambda y.\, \theta xy)\big) \;\leftrightarrow\; \Theta R$$

where (*sic*) $$\Theta \;\equiv\; \lambda H.\, H\big(\lambda x'y'.\, H(\lambda x''y''.\, \theta x'y'')\big),$$

so, since R is prime, $\Theta R \;\leftrightarrow\; R\big(\lambda xy.\, \Theta(\lambda\theta'.\, \theta' xy)\big) \;\leftrightarrow\; R\big(\lambda xy.\, \theta xy\big) \;\leftrightarrow\; R\theta$.

For the extended $\mathsf{focus}\,\eta$-rule, if $R \equiv \lambda\theta.\, \theta\langle x, y\rangle$ then

$$\begin{aligned}
\mathsf{focus}_{A\times B}\, R \;&\equiv\; \big\langle\, \mathsf{focus}_A\, \big(\lambda\phi.\, (\lambda\theta.\, \theta\langle x,y\rangle)(\lambda p.\, \phi(\pi_0 p))\big), \ldots \big\rangle \\
&\leftrightarrow\; \big\langle\, \mathsf{focus}_A\, \big(\lambda\phi.\, (\lambda p.\, \phi(\pi_0 p))\langle x,y\rangle\big), \ldots \big\rangle \\
&\leftrightarrow\; \big\langle\, \mathsf{focus}_A\, \big(\lambda\phi.\, \phi(\pi_0\langle x,y\rangle)\big), \ldots \big\rangle \\
&\leftrightarrow\; \big\langle\, \mathsf{focus}_A\, \big(\lambda\phi.\, \phi x\big), \ldots \big\rangle \;\leftrightarrow\; \langle x, y\rangle.
\end{aligned}$$

□

Lemma 8.8 $\mathsf{focus}_{\Sigma^{\vec{A}}}\, \mathcal{P} \;\equiv\; \lambda\vec{x}.\, \mathcal{P}(\lambda\phi.\, \phi\vec{x})$ (Lemma A 8.8).

Proof Consider $\mathbf{F}\Phi \equiv F\big(\lambda\vec{x}.\, \Phi(\lambda\phi.\, \phi\vec{x})\big)$ in the definition of primality of $\mathcal{P}$, so

$$\begin{aligned}
F(\mathsf{focus}_{\Sigma^{\vec{A}}}\, \mathcal{P}) \;&\equiv\; F\big(\lambda\vec{x}.\, \mathcal{P}(\lambda\phi.\, \phi\vec{x})\big) \;\equiv\; \mathbf{F}\mathcal{P} \\
&\leftrightarrow\; \mathcal{P}\big(\lambda\phi.\, \mathbf{F}(\lambda G.\, G\phi)\big) \;\leftrightarrow\; \mathcal{P}\big(\lambda\phi.\, F(\lambda\vec{x}.\, \phi\vec{x})\big) \;\equiv\; \mathcal{P}F,
\end{aligned}$$

which justifies the extended $\mathsf{focus}\,\beta$-rule and says how to apply F *to* a focus-term. For the extended $\mathsf{focus}\,\eta$-rule,

$$\begin{aligned}
\mathsf{focus}(\lambda F.\, F\phi) \;&\equiv\; \lambda\vec{x}.\, (\lambda F.\, F\phi)(\lambda\psi.\, \psi\vec{x}) \\
&\leftrightarrow\; \lambda\vec{x}.\, (\lambda\psi.\, \psi\vec{x})\phi \;\leftrightarrow\; \lambda\vec{x}.\, \phi\vec{x} \;\leftrightarrow\; \phi.
\end{aligned}$$

Application *of* this focus-term to a string of arguments is given by

$$(\mathsf{focus}\, \mathcal{P})\vec{a} \;\equiv\; (\lambda\phi.\, \phi\vec{a})(\mathsf{focus}\, \mathcal{P}) \;\equiv\; \mathcal{P}(\lambda\phi.\, \phi\vec{a}) \;\equiv\; \big(\lambda\vec{x}.\, \mathcal{P}(\lambda\phi.\, \phi\vec{x})\big)\vec{a}.$$

□

To summarise, the focus operation is only needed on the outside of the urterm, if at all:

Theorem 8.9 Any well formed term of the sober λ-calculus is interchangeable with
(a) σ if it is of urtype Σ;
(b) $\mathsf{focus}\, P$, if it is of base type;
(c) $\lambda\vec{x}.\, \sigma$, if it is of exponential urtype; or
(d) $\langle a, b\rangle$, if it is of product urtype;
where σ and P are urterms (without focus) and a and b are also of this form (Proposition A 8.10).
□

Remark 8.10 Generalising the cut rule to allow terms with focus is a lot more complicated because of the primality pre-condition and this problem will get worse in equideductive logic. Therefore, instead of asserting the more general form of cut as an *axiom*, we now *derive* the rule

$$\frac{\Gamma \;\vdash\; a : A \qquad\qquad x : A,\, \Delta \;\vdash\; b : B}{\Gamma,\, \Delta \quad\vdash\quad [a/x]^* b : B}$$

in the two cases where focus is the outermost operation of a or of b. We must then show that our definitions obey Axiom 5.5(c), transmitting the interchanges

$$\mathsf{focus}\, P \;\leftrightarrow\; \mathsf{focus}\, P', \quad a \;\leftrightarrow\; \mathsf{focus}\, P, \quad a \;\leftrightarrow a'$$

and similarly with b, b', Q and Q', but most of these follow from Lemma 8.4 and the fact that our formulae already respect non-focus interchanges.

The first case is not so much substitution *of* a focus term for occurrences of the variable x *inside* the term b as *around* it.

Lemma 8.11 $[\mathsf{focus}_A\, P/x]^*b \;\equiv\; \mathsf{focus}_B\, \big(\lambda\psi.\, P(\lambda x.\, \psi b)\big)$.

Proof We treat the term $x : A, \Delta \vdash b \equiv fx : B$ as a morphism $f : A \to B$ between exponentiable objects and use Lemma 7.3, which comes from naturality of η. Then we we may define

$$f(\mathsf{focus}_A\, P) \;\equiv\; [\mathsf{focus}_A\, P/x]^*(fx) \;\equiv\; \mathsf{focus}_B\, \big((\Sigma^{\Sigma^f})P\big)$$

because, in the urcontext $\Gamma, \Delta, \psi : \Sigma^B$,

$$\psi\big(\,\mathsf{focus}_B((\Sigma^{\Sigma^f})P)\big) \;\leftrightarrow\; ((\Sigma^{\Sigma^f})P)\psi \;\leftrightarrow\; P(\psi \cdot f),$$

which is what we require for $(\psi \cdot f)(\mathsf{focus}_A\, P)$. The one non-trivial interchange is $a \leftrightarrow \mathsf{focus}\, P$, from which we deduce $P \leftrightarrow \lambda\phi.\, \phi a$,

$$P(\lambda x.\, \psi b) \;\leftrightarrow\; (\lambda\phi.\, \phi a)(\lambda x.\, \psi b) \;\leftrightarrow\; (\lambda x.\, \psi b)a \;\leftrightarrow\; \psi[a/x]^*b$$

and so $\quad \mathsf{focus}\, \big(\lambda\psi.\, P(\lambda x.\, \psi b)\big) \;\leftrightarrow\; \mathsf{focus}\, \big(\lambda\psi.\, \psi([a/x]^*b)\big) \;\leftrightarrow\; [a/x]^*b.$ □

For subsitution *into* a focus term, recall from Remark 7.1 that a morphism is prime diagrammatically if it has equal composites with a certain outgoing pair, so any precomposition with it has the same property:

Lemma 8.12 $[a/x]^*(\mathsf{focus}_B\, Q) \;\equiv\; \mathsf{focus}_B\, \big([a/x^*Q]\big)$.

Proof If $x : A, \Delta \vdash Q : \Sigma^{\Sigma^B}$ is prime then so is $\Gamma, \Delta \vdash [a/x]^*Q : \Sigma^{\Sigma^B}$ because

$$\begin{aligned}\Phi([a/x]^*Q) \;\leftrightarrow\; [a/x]^*(\Phi Q) \quad &\leftrightarrow \quad [a/x]^*\big(Q(\lambda x.\, \Phi(\lambda\Phi.\, \phi x))\big) \\ &\leftrightarrow \quad \big([a/x]^*Q\big)\big(\lambda x.\, \Phi(\lambda\Phi.\, \phi x))\big)\end{aligned}$$

in the urcontext $\Gamma, \Delta, \Phi : \Sigma^3 B$, so the definition is legitimate. For the $\mathsf{focus}\,\beta$-rule,

$$\phi\big(\,\mathsf{focus}([a/x]^*Q)\big) \;\leftrightarrow\; \big([a/x]^*Q\big)\phi \;\leftrightarrow\; [a/x]^*(Q\phi) \;\leftrightarrow\; [a/x]^*\big(\phi(\mathsf{focus}\, Q)\big),$$

which is what we require for $\phi\big([a/x]^*(\mathsf{focus}\, Q)\big)$. The interchange $b \leftrightarrow \mathsf{focus}\, Q$ gives $Q \leftrightarrow \lambda\psi.\, \psi b$ and

$$[a/x]^*Q \;\leftrightarrow\; \lambda\psi.\, \psi\big([a/x]^*b\big) \quad \text{and so} \quad \mathsf{focus}\, \big([a/x]^*Q\big) \;\leftrightarrow\; [a/x]^*b.$$ □

Corollary 8.13 $[\mathsf{focus}\, P/a]^*(\mathsf{focus}\, Q) \;\leftrightarrow\; \mathsf{focus}\, \big(\lambda\psi.\, P(\lambda x.\, Q\psi)\big)$.

Proof This follows from Lemma 8.11 since $\psi(\mathsf{focus}\, Q) \leftrightarrow Q\psi$, but also from Lemmas 8.12 and 8.8 and $\lambda\beta$. □

Now we return from syntax to category theory.

Theorem 8.14 The sober λ-calculus may be interpreted in any equideductive category.

Proof The interpretation $\llbracket A \rrbracket$ of any urtype A is already required to be exponentiable. It is therefore sober by Proposition 7.7, so we have an equaliser diagram in $\mathcal{Q}$:

$$
\begin{tikzcd}
\llbracket \Gamma \rrbracket \arrow[d, dotted, "\llbracket \mathsf{focus}\, P \rrbracket"'] \arrow[drr, "\llbracket P \rrbracket"] & & \\
\llbracket A \rrbracket \arrow[rr, tail, "\eta_{\llbracket A \rrbracket}"] & & \Sigma^2 \llbracket A \rrbracket = \llbracket \Sigma^{\Sigma^A} \rrbracket \arrow[r, shift left, "\eta \Sigma^2 \llbracket A \rrbracket"] \arrow[r, shift right, "\Sigma^2 \eta \llbracket A \rrbracket"'] & \Sigma^4 \llbracket A \rrbracket = \llbracket \Sigma^4 A \rrbracket
\end{tikzcd}
$$

The interpretation $\llbracket P \rrbracket$ of any prime $\Gamma \vdash P : \Sigma^{\Sigma^A}$ is a map that has the same composite with the parallel pair, so it factors through the equaliser. The $\mathsf{focus}\,\beta$-rule is commutativity of the triangle. The interchange-transmitting rule ($\mathsf{focus} \leftrightarrow$) is valid because η is the inclusion of an equaliser and therefore mono. In the case $P \equiv \lambda\phi.\ \phi a$, the mediator is $\llbracket a \rrbracket$, so the η-rule is also valid. $\square$

Whilst we previously required all objects $\llbracket \Gamma \rrbracket$ of $\mathcal{Q}$ to respect exponentials and equalisers, we did not actually rely on this in this proof, although we have done in earlier results. This is because an urcontext Γ so far consists only of urtyped variables and no equations, so its interpretation $\llbracket \Gamma \rrbracket$ is an urspace. This is somewhat unnatural given that primality itself is an equation, so equideductive logic will allow such equations as hypotheses. The stronger requirement will then become relevant in Lemma 11.11.

Remark 8.15 The category of contexts and substitutions $\mathsf{Cn}^{\mathsf{sob}}_{\mathcal{L}}$ for the sober λ-calculus can be constructed in a similar way to Theorem 6.3. The objects (urcontexts) are the same, whilst the term language is enriched by the focus operation.

However, since focus is only needed on the outside of a term, another way to construct the category is as the opposite of the Kleisli category for the $\Sigma^{(-)} \dashv \Sigma^{(-)}$ monad. This is how it was done in [A] and essentially in [Füh99, Sel01, Thi97].

Theorem 8.16 Let $\mathcal{A}$ be a category that has products, $\Sigma^{(-)}$ and all objects sober. Then $\mathcal{A}$ is equivalent to $\mathsf{Cn}^{\mathsf{sob}}_{\mathcal{L}}$ for its proper language.

Proof By Theorem 6.8, $\mathcal{A} \simeq \mathsf{Cn}^{\lambda}_{\mathcal{L}} \to \mathsf{Cn}^{\mathsf{sob}}_{\mathcal{L}}$ for the proper restricted λ-calculus of $\mathcal{A}$, so we need to define the pseudo-inverse functor $\mathsf{Cn}^{\mathsf{sob}}_{\mathcal{L}} \to \mathsf{Cn}^{\lambda}_{\mathcal{L}}$. These two categories have the same objects and we just have to interpret the new operation $\mathsf{focus}_{\ulcorner A \urcorner}$, which recognises in the syntax the semantic sobriety of the object $A \in \mathcal{A}$. $\square$

This completes the discussion of the object language, so we are now ready to introduce equideductive predicates.

9 Equideductive logic

The symbolic language (pairing, λ-abstraction, application and focus) that we have introduced so far accounts for parts (b–e) of Definition 4.11 for an equideductive category. In this section we describe the new calculus that justifies Notation 1.1 for partial products (parts a and f), although we leave out focus until the next section.

Remark 9.1 First we put our new logic in the setting of type theories in general. It is a *two-level dependent type theory* that (like the many-sorted first order predicate calculus that it suggests) has a *division of contexts* into

(a) ***urtypes***, with no dependency, even amongst the urtypes themselves, and

(b) ***predicates***, depending on (variables that range over) urtypes but not predicates.

Urtypes are related by ***terms***. Although these are essentially those of the sober λ-calculus, we do not adopt the interchange rules that we stated before. This is because the burden of reasoning about equality of terms will instead be transferred on to the new theory of predicates. We show that the old axioms become theorems of the new calculus.

Predicates are similarly related by ***proofs***. Unlike terms, these are *anonymous* (or *irrelevant*), *i.e.* we do not distinguish between two proofs of the same predicate, at least as we study the calculus in this paper.

Axiom 9.2 In equideductive logic, *cf.* Axiom 5.2,

(a) the *urtypes* are the same as before (Axiom 5.2(a)), being generated from base types such as $\mathbf{0}$, $\mathbf{1}$ and $\mathbb{N}$ by $\times$ and $\Sigma^{(-)}$;

(b) a *context* is a list of distinct urtyped variables, *i.e.* an *ur*context in the sense of Axiom 5.2(b), *together with* a list of urpredicates, whose free variables are amongst those in the urcontext;

(c) the *terms* and *term-formation judgements* are those of the restricted λ-calculus (Axiom 5.6), to which we shall add focus in the next section;

(d) the *structural rules for variables* are as in Axiom 5.4, but since there may be predicates $\vec{\mathfrak{r}}(y, \vec{z}\,)$ in the contexts (on the left of $\vdash$ in a judgement as well as the right), these too undergo substitution in the cut rule (Axiom 9.7):

$$\frac{\vec{x}:\vec{A},\ \vec{\mathfrak{p}}(\vec{x}\,)\ \vdash\ b:B \qquad\qquad y:B,\ \vec{z}:\vec{C},\ \vec{\mathfrak{q}}(y),\ \vec{\mathfrak{r}}(y,\vec{z}\,)\ \vdash\ d:D}{\vec{x}:\vec{A},\ \vec{z}:\vec{C},\ \vec{\mathfrak{p}}(\vec{x}\,),\ \vec{\mathfrak{q}}(b),\ \vec{\mathfrak{r}}(b,\vec{z}\,)\ \vdash\ [b/y]^{*}d:D}$$

(Cut with predicates doesn't actually have any effect on urterms.)

The principal generalisation is from *interchanges* to *predicates*, which also have their own judgements and structural rules that we list below. Because of the difficulties that focus causes, in this section we define *ur*predicates without allowing it. When we do add it, in the next section, the generalisation is only a syntactic one because the terms that are involved are of type Σ, so it will be possible to eliminate focus from them.

It will be more important in equideductive logic than in the familiar predicate calculus to know exactly what the arguments of a predicate are, so we state them explicitly as $\vec{x}$. When we have proved the rules for the product we shall be able to use a single argument instead of a string. By a predicate "on" an urtype, we mean that this is the type of its argument.

Axiom 9.3 The ***urpredicates*** of equideductive logic are generated as follows:

(a) $\top$ is an urpredicate on any urtype;

(b) if $\mathfrak{p}(\vec{y}\,)$ is an urpredicate on $\vec{B}$ (*i.e.* with free variables $\vec{y}:\vec{B}$) and $\vec{x}:\vec{A} \vdash \vec{f}(\vec{x}\,):\vec{B}$ are urterms of the restricted λ-calculus then $\mathfrak{p}\big(\vec{f}(\vec{x}\,)\big)$ is an urpredicate on $\vec{A}$;

(c) in particular, considering $\vec{A}, \vec{B} \vdash \vec{A}$, any urpredicate $\mathfrak{p}(\vec{x}\,)$ on $\vec{A}$ is also $\mathfrak{p}(\vec{x}, \vec{y}\,)$ on $\vec{A}, \vec{B}$, so we may freely add arguments on which $\mathfrak{p}$ doesn't actually depend;

(d) if $\mathfrak{p}(\vec{x}\,)$ and $\mathfrak{q}(\vec{x}\,)$ are urpredicates on $\vec{A}$ then so is $\mathfrak{p}(\vec{x}\,) \,\&\, \mathfrak{q}(\vec{x}\,)$;

(e) if $\vec{\mathfrak{q}}(\vec{y}\,)$ are urpredicates on $\vec{B}$ (where the variables $\vec{y}$ are distinct from $\vec{x}$) and $\vec{x}:\vec{A}, \vec{y}:\vec{B} \vdash \alpha, \beta : \Sigma$ are urterms (not involving focus) then

$$\mathfrak{p}(\vec{x}\,)\ \equiv\ \forall \vec{y}{:}\vec{B}.\ \ \vec{\mathfrak{q}}(\vec{y}\,)\ \Longrightarrow\ (\alpha\vec{x}\vec{y} = \beta\vec{x}\vec{y}\,)$$

is an urpredicate on $\vec{A}$;

(f) we omit the symbol $\Longrightarrow$ if there are no ***antecedents*** $\vec{\mathfrak{q}}$ to put on the left of it; and

(g) we omit $\forall$ if there are no bound variables $\vec{y}$, but because of the variable binding rule we can only form an unquantified implication $\mathsf{q} \Longrightarrow \alpha\vec{x} = \beta\vec{x}$ when the antecedent q has *no* free variables;

(h) in particular, any equation $\mathfrak{p}(\vec{x}\,) \equiv (\alpha\vec{x} = \beta\vec{x}\,)$ between urterms of type Σ is an urpredicate on $\vec{A}$.

Example 9.4 For any urterm $\Gamma \vdash P : \Sigma^{\Sigma^A}$, we express Definition 7.2 in equideductive logic as the urpredicate

$$\mathsf{prime}(P) \quad \equiv \quad \forall \Phi{:}\Sigma^3 A.\ \ \Phi P \ = \ P\big(\lambda x.\, \Phi(\lambda\phi.\, \phi x)\big).$$

Remark 9.5 The use of $\forall\!\Longrightarrow$ is governed by the ***variable-binding rule*** (Definition 1.2): all of the variables that occur on the left of $\Longrightarrow$ must be bound by $\forall$. This means that $\Longrightarrow$ is much less than Heyting implication: it is really just conjunction indexed by $\{\vec{y} : \vec{B} \mid \vec{\mathfrak{p}}(\vec{y}\,)\}$. It is also a reason for stating the free variables in urpredicates explicitly. (In fact, the predicates with *no* free variables at all do form a Heyting algebra, including disjunction, as we shall see in [BB].)

Amongst the various operators, our convention is that application binds most tightly, followed by λ-abstraction, equality ($=$), conjunction ($\&$), disjunction ($\curlyvee$), implication ($\Longrightarrow$), quantification ($\forall, \ni$) and finally the turnstile ($\vdash$).

Remark 9.6 We shall introduce more general notation for predicates in Notation 9.10 and in later papers, but these will be *definitional extensions* that may always be put back into the form that we have described. An *ur*predicate is a conjunction of quantified implications, each of which has an equation of urtype Σ on the right of $\Longrightarrow$, and (recursively) a similar conjunction on the left. Therefore, in order to prove something for *all* predicates, we only need to consider the three cases

$$\top, \qquad \mathsf{p}(\vec{x}\,)\,\&\,\mathsf{q}(\vec{x}\,), \qquad \forall\vec{y}.\,\vec{\mathfrak{q}}(\vec{y}\,) \Longrightarrow (\alpha\vec{x}\vec{y} = \beta\vec{x}\vec{y}\,),$$

in which $\alpha\vec{x}\vec{y}$ and $\beta\vec{x}\vec{y}$ are urterms of type Σ, not involving focus. We shall sometimes treat (unquantified and unqualified) equations on their own as a fourth case.

The theory of the (restricted) λ-calculus in Section 5 was about interchanges between urterms, which could not be quantified and only appeared on the right of $\vdash$. In equideductive logic we have predicates instead of interchanges and replace the structural rules in Axiom 5.5 by

Axiom 9.7 The ***structural rules for urpredicates*** are

(a) reflexivity, symmetry and transitivity of equality;

(b) *hypothesis*: any urpredicate from the context, *i.e.* on the left of $\vdash$, may be asserted as a judgement, *i.e.* copied to the right of $\vdash$;

(c) *weakening*: any urpredicate whose free variables already belong to a context may be added to it;

(d) *exchange* and *contraction*: the urpredicates in a context may be permuted and repetitions may be deleted, so the list is just a set, *i.e.* we ignore order and multiplicity;

(e) *cut* of an urterm for a variable induces a substitution into the urpredicates on both sides of the $\vdash$:

$$\frac{\vec{x}:\vec{A},\ \vec{\mathfrak{p}}(\vec{x}\,)\ \vdash\ b:B \qquad\qquad y:B,\ \vec{z}:\vec{C},\ \vec{\mathfrak{q}}(y),\ \vec{\mathfrak{r}}(y,\vec{z}\,)\ \vdash\ \vec{\mathfrak{s}}(y,\vec{z}\,)}{\vec{x}:\vec{A},\ \vec{z}:\vec{C},\ \vec{\mathfrak{p}}(\vec{x}\,),\ \vec{\mathfrak{q}}(b),\ \vec{\mathfrak{r}}(b,\vec{z}\,)\ \vdash\ \vec{\mathfrak{s}}(b,\vec{z}\,)}$$

(f) *cut* for a predicate into another predicate:

$$\frac{\vec{x}:\vec{A},\ \vec{\mathfrak{p}}(\vec{x}\,)\ \vdash\ \mathsf{q}(\vec{x}\,) \qquad\qquad \vec{x}:\vec{A},\ \vec{z}:\vec{C},\ \mathsf{q}(\vec{x}\,),\ \vec{\mathfrak{r}}(\vec{x},\vec{z}\,)\ \vdash\ \vec{\mathfrak{s}}(x,\vec{z}\,)}{\vec{x}:\vec{A},\ \vec{z}:\vec{C},\ \vec{\mathfrak{p}}(\vec{x}\,),\ \vec{\mathfrak{r}}(x,\vec{z}\,)\ \vdash\ \vec{\mathfrak{s}}(x,\vec{z}\,)}$$

(g) *cut* for a predicate into term-formation,

$$\frac{\vec{x}:\vec{A},\ \vec{\mathfrak{p}}(\vec{x}\,)\ \vdash\ \mathfrak{q}(\vec{x}\,) \qquad\qquad \vec{x}:\vec{A},\ \vec{z}:\vec{C},\ \mathfrak{q}(\vec{x}\,),\ \vec{\mathfrak{r}}(\vec{x},\vec{z}\,)\ \vdash\ d:D}{\vec{x}:\vec{A},\ \vec{z}:\vec{C},\ \vec{\mathfrak{p}}(\vec{x}\,),\ \vec{\mathfrak{r}}(x,\vec{z}\,)\ \vdash\ d:D}$$

will only become relevant when we introduce focus.

The four versions of cut may be combined into one:

$$\frac{\vec{x}:\vec{A},\ \vec{\mathfrak{p}}(\vec{x}\,)\ \vdash\ \vec{b}:\vec{B},\ \vec{\mathfrak{q}}(\vec{b}) \qquad \vec{y}:\vec{B},\ \vec{z}:\vec{C},\ \vec{\mathfrak{q}}(\vec{y}\,),\ \vec{\mathfrak{r}}(\vec{y},\vec{z}\,)\ \vdash\ \vec{d}:\vec{D},\ \vec{\mathfrak{s}}(\vec{d}\,)}{\vec{x}:\vec{A},\ \vec{z}:\vec{C},\ \vec{\mathfrak{p}}(\vec{x}\,),\ \vec{\mathfrak{r}}(\vec{b},\vec{z}\,)\ \vdash\ [\vec{b}/\vec{y}]^*\vec{d}:\vec{D},\ \vec{\mathfrak{s}}([\vec{b}/\vec{y}]^*\vec{d}\,)}$$

Turning to the main features of the logic, we shall use cut wherever necessary, without comment. Issues such as cut elimination are not the subject of the present paper and we would like to keep the formality of the deduction to a minimum when we prove theorems about equideductive categories. So we shall blur the distinctions amongst right or introduction, and left or elimination rules, and those using variables or urterms. On the other hand, a future study of the *proof theory* of equideductive logic may yield interesting alternative models of it.

Axiom 9.8 The ***logical rules*** for $\top$ and & are

$$\Gamma\ \vdash\ \top \qquad\qquad \Gamma,\quad \mathfrak{p}(\vec{x}\,),\quad \mathfrak{q}(\vec{x}\,)\quad\vdash\quad \mathfrak{p}(\vec{x}\,)\ \&\ \mathfrak{q}(\vec{x}\,) \qquad\qquad \top I,\ \&I$$

$$\Gamma,\quad \mathfrak{p}(\vec{x}\,)\ \&\ \mathfrak{q}(\vec{x}\,)\quad\vdash\quad \mathfrak{p}(\vec{x}\,) \qquad\qquad \Gamma,\quad \mathfrak{p}(\vec{x}\,)\ \&\ \mathfrak{q}(\vec{x}\,)\quad\vdash\quad \mathfrak{q}(\vec{x}\,) \qquad\qquad \&E$$

or

$$\frac{\Gamma\ \vdash\ \mathfrak{r}(\vec{x}\,)}{\Gamma,\ \top\ \vdash\ \mathfrak{r}(\vec{x}\,)} \qquad\qquad \frac{\Gamma,\quad \mathfrak{p}(\vec{x}\,),\quad \mathfrak{q}(\vec{x}\,)\quad\vdash\quad \mathfrak{r}(\vec{x}\,)}{\Gamma,\quad \mathfrak{p}(\vec{x}\,)\ \&\ \mathfrak{q}(\vec{x}\,)\quad\vdash\quad \mathfrak{r}(\vec{x}\,)} \qquad\qquad \&I,\&E$$

Axiom 9.9 The ***logical rules*** for $\forall \Rightarrow$ are

$$\frac{\Gamma,\quad \vec{y}:\vec{A},\quad \vec{\mathfrak{p}}(\vec{y}\,)\quad\vdash\quad \alpha\vec{y}\ =\ \beta\vec{y}}{\Gamma\quad\vdash\quad \forall\vec{y}{:}\vec{A}.\ \vec{\mathfrak{p}}(\vec{y}\,)\ \Longrightarrow\ \alpha\vec{y}\ =\ \beta\vec{y}} \qquad\qquad \forall I,\forall E$$

Beware that, in order to satisfy the variable-binding rule, the $\vec{\mathfrak{p}}(\vec{y}\,)$ must not depend on any of the variables in Γ, although α and β may do so. Conversely, since the variables $\vec{y}$ are bound, *all* of the predicates on the left of $\vdash$ that depend on them must be moved to the left of $\Rightarrow$.

The upward part of this two-way $\forall I$ rule is $\forall E$ for variables ($a \equiv y$); this is the most convenient formulation for showing how to interpret the logic in a category (Proposition 11.9). We recover the $\forall E$-rule for urterms from it using cut:

$$\frac{\Gamma\ \vdash\ \vec{a}:\vec{A} \qquad \Gamma\ \vdash\ \vec{\mathfrak{p}}(\vec{a}\,) \qquad \Gamma\ \vdash\ \forall\vec{y}{:}\vec{A}.\ \vec{\mathfrak{p}}(\vec{y}\,)\Longrightarrow \alpha\vec{y}\ =\ \beta\vec{y}}{\Gamma\ \vdash\ \alpha\vec{a}\ =\ \beta\vec{a}} \qquad\qquad \forall E$$

Another useful form of the elimination rule is

$$\Gamma,\quad \mathfrak{p}(\vec{a}\,),\quad \forall\vec{y}.\ \mathfrak{p}(\vec{y}\,)\Longrightarrow \phi\vec{y}=\psi\vec{y}\quad\vdash\quad \phi\vec{a}=\psi\vec{a}. \qquad\qquad \forall E$$

As our first *definitional extension*, we can generalise implication to allow general predicates instead of just equations on the right of $\Rightarrow$.

Notation 9.10 $\forall y.\ \mathfrak{p}(y) \Rightarrow \mathfrak{q}(x,y)$ is defined as

$$\begin{array}{rll} \top & \text{if} & \mathfrak{q}(x,y)\equiv\top \\ \big(\forall y.\ \mathfrak{p}(y)\Rightarrow\mathfrak{r}(x,y)\big)\ \&\ \big(\forall y.\ \mathfrak{p}(y)\Rightarrow\mathfrak{s}(x,y)\big) & \text{if} & \mathfrak{q}(x,y)\equiv\mathfrak{r}(x,y)\ \&\ \mathfrak{s}(x,y) \\ \forall y.\ \forall z.\ \mathfrak{p}(y)\ \&\ \mathfrak{r}(z)\Rightarrow\alpha xyz=\beta xyz & \text{if} & \mathfrak{q}(x,y)\equiv\forall z.\ \mathfrak{r}(z)\Rightarrow\alpha xyz=\beta xyz. \end{array}$$

Proposition 9.11 These definitions satisfy the introduction and elimination rules in Definition 9.9, but with general predicates on the right.

Proof By the $\top$ rule and weakening, $(\forall y.\, \mathfrak{p}(y) \Rightarrow \top) \dashv\vdash \top$. For conjunction,

$$\begin{array}{c}
\Gamma,\ y:B,\ \mathfrak{p}(y)\ \vdash\ \mathfrak{r}(x,y)\,\&\,\mathfrak{s}(x,y) \\ \hline\hline
\begin{array}{cc}
\Gamma,\ y:B,\ \mathfrak{p}(y)\ \vdash\ \mathfrak{r}(x,y) & \Gamma,\ y:B,\ \mathfrak{p}(y)\ \vdash\ \mathfrak{s}(x,y) \\ \hline\hline
\Gamma\ \vdash\ \forall y.\, \mathfrak{p}(y) \Rightarrow \mathfrak{r}(x,y) & \Gamma\ \vdash\ \forall y.\, \mathfrak{p}(y) \Rightarrow \mathfrak{s}(x,y)
\end{array} \\ \hline\hline
\Gamma\ \vdash\ \big(\forall y.\, \mathfrak{p}(y) \Rightarrow \mathfrak{r}(x,y)\big)\ \&\ \big(\forall y.\, \mathfrak{p}(y) \Rightarrow \mathfrak{s}(x,y)\big)
\end{array}$$

For nested implication,

$$\begin{array}{c}
\Gamma,\quad y:B,\quad \mathfrak{p}(y)\quad \vdash\quad \forall z.\, \mathfrak{r}(z) \Rightarrow \alpha xyz = \beta xyz \\ \hline\hline
\Gamma,\quad y:B,\quad \mathfrak{p}(y),\quad z:C,\quad \mathfrak{r}(z)\quad \vdash\quad \alpha xyz = \beta xyz \\ \hline\hline
\Gamma,\quad \langle y,z\rangle : B\times C,\quad \mathfrak{p}(y)\ \&\ \mathfrak{r}(z)\quad \vdash\quad \alpha xyz = \beta xyz \\ \hline\hline
\Gamma\quad \vdash\quad \forall yz.\, \mathfrak{p}(y)\ \&\ \mathfrak{r}(z) \Rightarrow \alpha xyz = \beta xyz
\end{array}$$

This shows why we needed multiple quantifiers and hypotheses in Definition 9.3(e). $\square$

Turning to the rules for λ-terms, recall from Remark 5.1 that there are different ways of formulating those for equality. We used one of these in the restricted λ-calculus (Axiom 5.10), but in equideductive logic the other is more appropriate:

Axiom 9.12 Exponentials satisfy their β- and extensionality rules:

$$\frac{\Gamma\quad \vdash\quad \vec{a}:\vec{A} \qquad\qquad \Gamma,\ \vec{x}:\vec{A}\quad \vdash\quad \sigma:\Sigma}{\Gamma\quad \vdash\quad (\lambda\vec{x}.\,\sigma)\vec{a}\ =\ [\vec{a}/\vec{x}]^*\sigma} \qquad \lambda\beta$$

and
$$\phi,\psi:\Sigma^{\vec{A}},\quad F:\Sigma^{\Sigma^{\vec{A}}},\quad \forall\vec{x}.\,\phi\vec{x} = \psi\vec{x}\quad \vdash\quad F\phi = F\psi. \qquad \lambda\text{-ext}$$

Whereas extensionality for $\times$ said that π_0 and π_1 are jointly mono, for λ it says the same for the *family* of maps $\{\mathsf{ev}_{\vec{x}} : \Sigma^{\Pi\vec{A}} \to \Sigma \mid \vec{x}:\vec{A}\}$.

We use exponentials and a principle that is attributed to Gottfried Leibniz to extend the definition of equality from Σ to general urtypes:

Definition 9.13 ***Equality*** of terms of any urtype,

$$a = b : A, \qquad \text{is defined as} \qquad \forall\phi{:}\Sigma^A.\ \phi a = \phi b, \qquad \mathrm{T}_0$$

for which reflexivity, symmetry and transitivity follow from Axiom 9.7(a), together with $\forall E/I$. Since equality at general urtypes is defined by quantification, Proposition 9.11 allows us to use it on the right of $\Rightarrow$, making another definitional extension of the notation for predicates.

The remaining rules of the restricted λ-calculus follow:

Lemma 9.14 Equality for $\phi,\psi:\Sigma^A$ satisfies

$$\phi = \psi \quad \dashv\vdash \quad \forall F.\, F\phi = F\psi \quad \dashv\vdash \quad \forall x{:}A.\ \phi x = \psi x \qquad (*)$$

and the equality-transmitting and η-rules for the restricted λ-calculus (*cf.* Axiom 5.10).

Proof The forward direction of $(*)$ uses $\forall E$ with $F \equiv \lambda\theta.\,\theta x$ and the converse is λ-ext. Then this rule gives

$$\phi = \psi : \Sigma^A,\quad x:A\quad \vdash\quad \phi x = \psi x : \Sigma, \qquad \lambda E{=}_0$$

whilst $$\phi : \Sigma^A, \quad a = b : A \quad \vdash \quad \phi a = \phi b : \Sigma \qquad \lambda E{=}_1$$

comes from the Leibnizian definition of $a = b : A$, with $\forall E$. Then

$$\Gamma, \quad \phi : \Sigma^A \quad \vdash \quad \phi = \lambda a.\, \phi a : \Sigma^A \qquad \lambda\eta$$

is $\forall a'.\, \phi a' = (\lambda a.\, \phi a)a'$ by $\lambda\beta$ and $(*)$ again. Finally, in the rule

$$\frac{\Gamma,\ \vec{x} : \vec{A} \quad \vdash \quad \sigma = \tau : \Sigma}{\Gamma \quad \vdash \quad (\lambda\vec{x}.\, \sigma) \ = \ (\lambda\vec{x}.\, \tau)} \qquad \lambda I{=}$$

the top line is equivalent to $\forall x.\, (\lambda x.\, \sigma)x = (\lambda x.\, \tau)x$ by $\lambda\eta$ and $\forall I$. This is the same as the bottom line by $(*)$. Again we use cut to recover forms similar to Axiom 5.10. $\square$

Proposition 9.15 $\Gamma,\ \mathfrak{p}(a),\ a = b \ \vdash\ \mathfrak{p}(b)$.

Proof By Remark 9.6, we must consider the three cases in which $\mathfrak{p}$ is $\top$ (trivial), a conjunction (take the parts separately) or a quantified implication. The last follows from

$$\Gamma, \quad \forall y'.\, \mathfrak{q}(y') \Rightarrow \alpha a y' = \beta a y', \quad a = b, \quad \mathfrak{q}(y) \quad \vdash \quad \alpha b y = \beta b y$$

using $\forall E$, $\lambda E{=}_1$ and then $\lambda E{=}_0$. $\square$

Axiom 9.16 Products satisfy their β- and extensionality rules,

$$x : A,\ y : B \quad \vdash \quad \pi_0\langle x, y\rangle = x : A, \qquad \pi_1\langle x, y\rangle = y : B \qquad \times\beta$$

and $$p, q : A \times B, \quad \pi_0 p = \pi_0 q : A, \quad \pi_1 p = \pi_1 q : B \quad \vdash \quad p = q : A \times B, \qquad \times\text{-ext}$$

as in Axiom 5.8. For the nullary product we only need to say that

$$x : \mathbf{1} \quad \vdash \quad x = \star : \mathbf{1}. \qquad \mathbf{1}\text{-ext}$$

Lemma 9.17 The product $A \times B$ also satisfies its equality-transmitting and η-rules.

Proof Putting $\theta \equiv \phi \cdot \pi_0$ in Definition 9.13 for equality gives $\times E_0{=}$,

$$p = q : X \times Y \ \dashv\vdash\ \forall\theta.\, \theta p = \theta q \ \vdash\ \forall\phi.\, \phi(\pi_0 p) = \phi(\pi_0 q) \ \dashv\vdash\ \pi_0 p = \pi_0 q,$$

and similarly $\pi_1 p = \pi_1 q$. The proofs of $\times I{=}$ and $\times\eta$ are the same as in Lemma 5.9, but with equality $(=)$ instead of interchangeability $(\leftrightarrow)$. $\square$

Remark 9.18 In the restricted λ-calculus we made allowance for additional base types, operation-symbols and particular interchanges (Remark 5.3). In equideductive logic, there may also be ***particular axioms***, instead of just equations. These may be expressed in any of the following forms:

(a) equations $\Gamma \vdash \sigma = \tau$ of type Σ, possibly with equideductive premises,

(b) equations $\Gamma \vdash a = b$ of any type A,

(c) closed, unconditional predicates $\vdash \mathfrak{p}()$, or

(d) general judgements, $\vec{x} : \vec{A},\ \vec{\mathfrak{p}}(\vec{x}\,) \vdash \mathfrak{q}(\vec{x}\,)$,

which are inter-provable using the rules that we have stated in this section. The particular interchanges of the restricted λ-calculus are examples of (b) where Γ is an urcontext (with no predicates). In equideductive logic, on the other hand, we prefer to take (a) as canonical. Whereas in Definition 6.6 we took particular interchanges to be at base urtypes A, we now prefer just to use Σ, interpreting the previous convention by means of Leibnizian equality and an additional variable $\phi : \Sigma^A$ in the context. In Section 13 we shall use particular axioms to force the logic to match any given equideductive category.

10 Sobriety in equideductive logic

We now introduce focus into equideductive logic as we did for the restricted λ-calculus in Section 8 and then show how to eliminate the proof-theoretic difficulties that it causes.

Axiom 10.1 Sobriety is expressed by the introduction and β-rules

$$\frac{\Gamma \ \vdash \ P : \Sigma^{\Sigma^A} \qquad \Gamma \ \vdash \ \mathsf{prime}(P)}{\Gamma \ \vdash \ \mathsf{focus}_A\, P : A} \qquad\qquad \frac{\Gamma \ \vdash \ P : \Sigma^{\Sigma^A} \qquad \Gamma \ \vdash \ \mathsf{prime}(P)}{\Gamma,\ \phi : \Sigma^A \ \vdash \ \phi(\mathsf{focus}_A\, P) = P\phi}$$

for any *base* type A, where the predicate $\mathsf{prime}(P)$ was defined in Example 9.4. We have already stated the T_0 rule as Definition 9.13. To these we add pairing and the structural rules apart from cut, as before.

Since equideductive logic allows hypotheses in the context, unlike the sober λ-calculus in Section 8, an urterm may now be ***conditionally prime***, *i.e.* primality can depend on such hypotheses. It may also be combined with other equideductive predicates.

Instead of giving axioms for predicates involving focus, we shall show that they may be obtained as a definition extension, as we did when defining focus for product and exponential types. We also define cuts and substitutions of focus-terms into predicates.

The lemmas that did this in Section 8 remain valid with equideductive equality in place of interchangeability and when primality is allowed to be conditional on hypotheses.

.

Lemma 10.2 For any $\Gamma \vdash a : A$, the term $\Gamma \vdash P \equiv \lambda\phi.\ \phi a$ satisfies $\Gamma \vdash \mathsf{prime}(P)$.

Proof By $\lambda\beta$, in the context $\Gamma,\ \Phi : \Sigma^3 A$,

$$P\big(\lambda x.\ \Phi(\lambda\psi.\ \psi x)\big) \ \equiv\ (\lambda\phi.\ \phi a)\big(\lambda x.\ \Phi(\lambda\psi.\ \psi x)\big) \ =\ \Phi(\lambda\psi.\ \psi a) \ \equiv\ \Phi P,$$

so $\ \Gamma \ \vdash \forall\Phi.\ \Phi P = P\big(\lambda x.\ \Phi(\lambda\psi.\ \psi x)\big)\ $ by Axiom 9.9. □

Lemma 10.3 Sobriety obeys its η- and equality-transmitting laws (Axiom 8.1).

Proof The η-rule

$$x : A \ \vdash \ \mathsf{focus}_A(\lambda\phi.\ \phi x) \ =\ x : A, \qquad \mathsf{focus}\,\eta$$

which is well formed by the previous lemma, follows from the $\mathsf{focus}\,\beta$- and $\lambda\beta$-rules,

$$x : A,\ \theta : \Sigma^A \ \vdash \ \theta\big(\mathsf{focus}_A(\lambda\phi.\ \phi x)\big) \ =\ (\lambda\phi.\ \phi x)\theta \ =\ \theta x,$$

and Definition 9.13 for equality at urtype A. The equality-transmitting rule for focus is

$$P = Q \ \dashv\vdash\ \forall\phi.\ P\phi = Q\phi \ \dashv\vdash\ \forall\phi.\ \phi(\mathsf{focus}\, P) = \phi(\mathsf{focus}\, Q) \ \dashv\vdash\ \mathsf{focus}\, P = \mathsf{focus}\, Q, \qquad \mathsf{focus} =$$

which comes from Lemma 9.14, $\mathsf{focus}\,\beta$ and the definition of equality, *cf.* Lemma 8.4. □

Theorem 10.4 Any two terms that are definable and provably interchangeable in the sober λ-calculus obey the equality predicate in equideductive logic. □

The cut rule in the λ-calculus supplies a term to be substituted for a free variable in the context, but in a predicate calculus like equideductive logic it may also provide a proof for a hypothesis (Axiom 9.7(f,g)). We must therefore show how to define or eliminate the new cuts that assert predicates. However, there is only one of these in Axiom 10.1:

Lemma 10.5 The equation $\ \phi(\mathsf{focus}\, P) \ =\ P\phi\ $ is redundant as a hypothesis in the contexts of a proof, so it may simply be deleted from them.

Proof This equation is only well formed according to Definition 10.15 if $\mathsf{prime}(P)$ is already provable from the earlier part of the context in which it appears. It ought therefore to be redundant because it is an axiom ($\mathsf{focus}\,\beta$).

Suppose that we simply delete this equation wherever it occurs as a hypothesis in the contexts of a proof. All of the steps of the proof remain valid, with the exception of any instances of the hypothesis rule (Axiom 9.7(b)) that copy this equation from the left to the right of $\vdash$. These steps may be replaced by copies of the proof of $\mathsf{prime}(P)$ in the appropriate context, followed by the $\mathsf{focus}\,\beta$-rule. □

This completes the definitional extension of allowing the focus operation:

Theorem 10.6 The contexts and predicates in Definition 10.15 (possibly involving focus) satisfy the structural and logical rules that were given for urpredicates (without it) in the previous section. □

When we define the classifying category $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ in Section 12 we will need to identify its injective objects. Their syntactic characterisation will be as follows:

Definition 10.7 An urtype B is called ***syntactically injective*** if, whenever a term

$$\vec{x} : \vec{A}, \quad \vec{\mathfrak{p}}(\vec{x}\,) \quad \vdash \quad b : B$$

is provably well formed in equideductive logic (possibly using focus) there is some urterm (without focus),

$$\vec{x} : \vec{A} \quad \vdash \quad c : B,$$

that is already provably well formed in the restricted λ-calculus (Section 5) and the equation

$$\vec{x} : \vec{A}, \quad \vec{\mathfrak{p}}(\vec{x}\,) \quad \vdash \quad b = c : B$$

is provable in equideductive logic.

Lemma 10.8 The type Σ is syntactically injective.

Proof Let $\vec{x} : \vec{A},\ \vec{\mathfrak{p}}(\vec{x}\,) \ \vdash\ \sigma : \Sigma$ be a term that may involve focus. We may eliminate this by the normalisation theorem, possibly relying on the hypotheses $\vec{\mathfrak{p}}(\vec{x})$, to obtain an urterm τ. This may contain the variables $\vec{x} : \vec{A}$ but is unconditionally well formed. It is equal to σ under the given hypotheses. □

Lemma 10.9 Any exponential Σ^B is syntactically injective. Any product or retract of syntactically injective types is again syntactically injective. □

Proposition 10.10 The following are equivalent for an urtype A:
(a) A is syntactically injective;
(b) focus_A is representable by an urterm $F : \Sigma^{\Sigma^A} \vdash a_F : A$;
(c) A is a retract of some exponential Σ^B.

Proof [a⊢b]: The syntactic injectivity property applied to

$$P : \Sigma^{\Sigma^A}, \quad \mathsf{prime}(P) \quad \vdash \quad \mathsf{focus}\,P : A$$

yields an urterm $\quad F : \Sigma^{\Sigma^A} \ \vdash\ a_F : A \quad$ such that

$$P : \Sigma^{\Sigma^A},\ \mathsf{prime}(P) \ \vdash\ a_P \ =\ \mathsf{focus}\,P,$$

so $P\phi \;=\; \phi(\mathsf{focus}\,P) \;=\; \phi(a_P)$. Beware that, although one can deduce from this result that syntactically injective types are exponentiable, whilst F is a free variable in the term a_F, the restricted λ-calculus does not allow us to form $\lambda F.\, a_F$.

[b⊢c]: This term makes $A \lhd \Sigma^{\Sigma^A}$. [c⊢a]: By the previous result. □

Any urtype A may therefore be embedded in a syntactically injective object, namely Σ^{Σ^A}, but if we are going to use this instead of A itself then we also have to replace predicates on A by those on Σ^{Σ^A}.

Notation 10.11 For $F : \Sigma^{\Sigma^A}$, write $\quad \bar{\mathfrak{p}}(F) \;\equiv\; \forall \phi\psi.\, (\forall x.\, \mathfrak{p}(x) \Rightarrow \phi x = \psi x) \Longrightarrow F\phi = F\psi.$

Lemma 10.12 This satisfies a "***double negation***" property:

$$x : A,\ \mathfrak{p}(x) \;\dashv\vdash\; \bar{\mathfrak{p}}(\lambda\phi.\, \phi x) \;\equiv\; \forall \phi\psi.\, \big(\forall x'.\, \mathfrak{p}(x') \Rightarrow \phi x' = \psi x'\big) \Longrightarrow \phi x = \psi x.$$

Proof The forward direction is essentially $\forall E$. The most complicated case of the converse is

$$\mathfrak{p}(x) \;\equiv\; \forall y.\, \mathfrak{q}(y) \Rightarrow \alpha x y = \beta x y.$$

Using $\forall E$,

$$x' : A, \quad y : B, \quad \mathfrak{q}(y), \quad \forall y'.\, \mathfrak{q}(y') \Rightarrow \alpha x' y' = \beta x' y' \quad \vdash \quad \alpha x' y = \beta x' y,$$

so by $\forall I$,

$$y : B, \quad \mathfrak{q}(y) \quad \vdash \quad \forall x'.\, \big(\forall y'.\, \mathfrak{q}(y') \Rightarrow \alpha x' y' = \beta x' y'\big) \Longrightarrow \alpha x' y = \beta x' y,$$

which is $\qquad y : B, \quad \mathfrak{q}(y) \quad \vdash \quad \forall x'.\, \mathfrak{p}(x') \Rightarrow \alpha x' y = \beta x' y.$

Together with this, $\phi \equiv \lambda x'.\, \alpha x' y$ and $\psi \equiv \lambda x'.\, \beta x' y$ in $\bar{\mathfrak{p}}(\lambda\theta.\, \theta x)$ give

$$x : A, \quad \bar{\mathfrak{p}}(\lambda\theta.\, \theta x), \quad y : B, \quad \mathfrak{q}(y) \quad \vdash \quad \phi x = \alpha x y = \beta x y = \psi y,$$

so $\qquad x : A, \quad \bar{\mathfrak{p}}(\lambda\theta.\, \theta x) \quad \vdash \quad \big(\forall y.\, \mathfrak{q}(y) \Rightarrow \alpha x y = \beta x y\big) \quad \equiv \quad \mathfrak{p}(x).$

The case of $\mathfrak{p}(x) \equiv \top$ is $\forall E$. If $\mathfrak{p}(x) \equiv \mathfrak{q}(x) \,\&\, \mathfrak{r}(x)$ then $\mathfrak{p}(x) \vdash \mathfrak{q}(x)$ so $\bar{\mathfrak{p}}(\lambda\theta.\, \theta x) \vdash \bar{\mathfrak{q}}(\lambda\theta.\, \theta x) \vdash \mathfrak{q}(x)$. These exhaust the possibilities by Remark 9.6. □

Warning 10.13 This can only be done if the sub-formula $\mathfrak{q}$ does not depend on x, so the variable-binding rule (Definition 1.2) is essential.

Hence we may embed any context in an injective object like this:

Lemma 10.14 For any term $\vec{x} : \vec{A},\ \vec{\mathfrak{p}}(\vec{x}\,) \vdash b : B,\ \mathfrak{q}(b)$ in equideductive logic, there is an urterm

$$\vec{x} : \vec{A} \;\vdash\; (Q\vec{x}\,) : \Sigma^{\Sigma^B}$$

that is definable in the restricted λ-calculus and satisfies

$$\vec{x} : \vec{A},\ \vec{\mathfrak{p}}(\vec{x}\,) \quad \vdash \quad (Q\vec{x} = \lambda\psi.\, \psi b), \quad \mathsf{prime}(Q\vec{x}\,), \quad \bar{\mathfrak{q}}(Q\vec{x}\,)$$

in equideductive logic. Conversely, every such urterm

$$\vec{x} : \vec{A} \;\vdash\; Q\vec{x} : \Sigma^{\Sigma^A} \qquad \text{such that} \qquad \vec{x} : \vec{A},\ \vec{\mathfrak{p}}(\vec{x}\,) \;\vdash\; \mathsf{prime}(Q\vec{x}\,), \quad \bar{\mathfrak{q}}(Q\vec{x}\,)$$

arises in this way from a term $b \equiv \mathsf{focus}(Q\vec{x}\,)$ with $\mathfrak{q}(b)$.

The (ur)terms Q and b are unique in the sense that any alternatives Q' and b' satisfy

$$\vec{x} : \vec{A},\quad \vec{\mathfrak{p}}(\vec{x}\,) \quad\vdash\quad Q\vec{x} \;=\; Q'\vec{x} : \Sigma^{\Sigma^B} \quad\text{or}\quad b \;=\; b' : B.$$

Proof By Example 10.9(d) there is an urterm $(Q\vec{x}\,)$ that is defined without hypotheses and is conditionally equal to $\lambda\psi.\,\psi b$. Hence it is conditionally prime by Lemma 10.2 and Proposition 9.15. It satisfies $\vec{\mathfrak{q}}(Q\vec{x}\,)$ by Lemma 10.12.

Conversely, if $\vec{x} : \vec{A}, \mathfrak{p}(\vec{x}\,) \vdash \mathsf{prime}(Q\vec{x}\,)$ then

$$\vec{x} : A,\ \mathfrak{p}(\vec{x}\,) \quad\vdash\quad b \;\equiv\; \mathsf{focus}(Q\vec{x}\,) : B$$

is well formed by Axiom 10.1 and satisfies

$$\vec{x} : A,\ \mathfrak{p}(\vec{x}\,) \quad\vdash\quad Q\vec{x} \;=\; \lambda\phi.\,\phi b, \quad \mathfrak{q}(b)$$

by $\mathsf{focus}\,\beta$ and Lemma 10.12. It is unique by Lemma 10.3. $\square$

Definition 10.15 ***Predicates*** and ***contexts*** possibly containing focus are defined by the following rules:

$$\frac{\Gamma,\quad \vec{y} : \vec{B},\quad \vec{\mathfrak{q}}(\vec{y}\,) \quad\vdash\quad \sigma, \tau : \Sigma}{\Gamma \quad\vdash\quad \forall\vec{y}.\,\vec{\mathfrak{q}}(\vec{y}\,) \Longrightarrow \sigma = \tau \ \ \mathsf{pred}}$$

where the proofs that the terms σ and τ are well formed may involve applying focus to other terms that are prime conditionally on $\vec{\mathfrak{q}}(\vec{y}\,)$, although of course we also allow the possibility that this sequence be empty; and

$$\frac{\Gamma\ \ \mathsf{ctxt} \quad x \notin \Gamma \quad A\ \ \mathsf{type}}{[\Gamma,\ x : A]\ \ \mathsf{ctxt}} \qquad\qquad \frac{\Gamma \quad\vdash\quad \mathfrak{p}\ \ \mathsf{pred}}{[\Gamma,\ \mathfrak{p}]\ \ \mathsf{ctxt}}$$

Informally, for any term, predicate or context involving $\mathsf{focus}\,P$ to be well formed, $\mathsf{prime}(P)$ must be provable from the hypotheses to its left, in which we include the (quantified) antecedents of $\Longrightarrow$. (The exchange rule has to be restricted to make this meaningful.)

This more general definition of predicate must be accompanied by corresponding generalisations of the $\forall$-rules. On the other hand, we may have inserted focus into an urpredicate by a (generalised) cut or substitution for a variable. We therefore have to show that these two proof rules commute, or rather define the first in terms of the second:

Lemma 10.16 If P is prime and $\mathfrak{p}(x) \;\equiv\; \forall\vec{y}.\,\vec{q}(\vec{y}\,) \Longrightarrow \alpha\vec{y}x = \beta\vec{y}x$ then the predicate

$$\forall\vec{y}.\,\vec{q}(\vec{y}\,) \Longrightarrow P(\lambda x.\,\alpha\vec{y}x) = P(\lambda x.\,\beta\vec{y}x)$$

serves for $\mathfrak{p}(\mathsf{focus}\,P)$.

Proof The cut that is needed to define this,

$$\frac{\dfrac{\Gamma \quad\vdash\quad P : \Sigma^{\Sigma^A},\ \mathsf{prime}(P)}{\Gamma \quad\vdash\quad \mathsf{focus}\,P : A} \qquad \begin{array}{c} x : A,\ \Delta,\ \vec{y} : \vec{B},\ \vec{q}(\vec{y}\,) \quad\vdash\quad \alpha\vec{y}x = \beta\vec{y}x \\ \hline\hline x : A,\ \Delta \quad\vdash\quad \mathfrak{p}(x) \;\equiv\; \forall\vec{y}{:}\vec{B}.\,\vec{q}(\vec{y}\,) \Longrightarrow \alpha\vec{y}x = \beta\vec{y}x \end{array}}{\Gamma,\ \Delta \quad\vdash\quad \mathfrak{p}(\mathsf{focus}\,P)}$$

is implemented by

$$\frac{\dfrac{\Gamma \quad\vdash\quad P : \Sigma^{\Sigma^A} \qquad \begin{array}{c} x : A,\ \Delta,\ \vec{y} : \vec{B},\ \vec{q}(\vec{y}\,) \quad\vdash\quad \alpha\vec{y}x = \beta\vec{y}x \\ \hline\hline x : A,\ \Delta,\ \vec{q}(\vec{y}\,) \quad\vdash\quad \lambda\vec{y}.\,\alpha\vec{y}x = \lambda\vec{y}.\,\beta\vec{y}x \end{array}}{\Gamma,\ \Delta, \vec{q}(\vec{y}\,) \quad\vdash\quad P(\lambda\vec{y}.\,\alpha\vec{y}x) = P(\lambda\vec{y}.\,\beta\vec{y}x)}}{\Gamma,\ \Delta \quad\vdash\quad \forall\vec{y}.\,\vec{q}(\vec{y}\,) \Longrightarrow P(\lambda x.\,\alpha\vec{y}x) = P(\lambda x.\,\beta\vec{y}x)}$$

$\square$

11 Interpretation in an equideductive category

Now we return to the category theory to show that our new logic can be interpreted in it, following the plan that we set out in Section 6. As before, we assume given an interpretation of the base types and operation-symbols of a language $\mathcal{L}$ in the subcategory $\mathcal{A} \subset \mathcal{Q}$ of urspaces, to which we add some particular axioms. We only need to interpret the minimal version of the logic that we set out in Section 9, but we shall also describe the behaviour of focus, which we introduced as a definitional extension in the previous section.

Remark 11.1 We may adopt parts (a–l) of Proposition 6.1 directly in order to interpret Axiom 9.2, that is,
(a) urtypes,
(b) urcontexts (lists of urtypes, without predicates),
(c) urterms (without focus) and
(d) the structural rules for variables.
In particular, the urtypes are interpreted as urspaces in $\mathcal{A} \subset \mathcal{Q}$ as before. Recall that these have to be exponentiable and therefore sober in that category.

Lemma 11.2 The interpretation $[\![A]\!]$ of any syntactically injective urtype A of $\mathcal{L}_\forall$ must be an injective object of $\mathcal{Q}$.

Proof By Proposition 10.10, A is a retract of an exponential urtype, so $[\![A]\!]$ is a retract of an exponential urspace of $\mathcal{Q}$, which is injective by Lemma 4.5. □

By the remaining parts of Proposition 6.1, we still have

Lemma 11.3 If two urterms $\vec{x} : \vec{A} \vdash b, c : B$ are interchangeable in the sense of the restricted λ-calculus then they are denoted by the same morphism $[\![b]\!] = [\![c]\!] : \Pi[\![\vec{A}]\!] \rightrightarrows [\![B]\!]$ in $\mathcal{Q}$. □

However, interchangeability of urterms in the restricted λ-calculus is not the same thing as obeying an equality predicate in equideductive logic. We therefore need to convert the interpretation of one calculus into the other. We begin with those things that can be done using finite limits (products, pullbacks and equalisers) in the category $\mathcal{Q}$.

Lemma 11.4 The intepretation takes any syntactic product cone of urtypes

$$A \xleftarrow{\pi_0} A \times B \xrightarrow{\pi_1} B$$

to a categorical product of urspaces in $\mathcal{A} \subset \mathcal{Q}$. Moreover, maps from any object of $\mathcal{Q}$ to this product respect its universal property. Hence the rules for $\times$ stated in Axiom 9.16 and Lemma 9.17 are sound, *i.e.* if two urterms satisfy the equality predicate then they have the same denotation, without invoking the proof of this Lemma. □

Next, recall from Section 9 that an *ur*predicate is defined in an urcontext (a list of variables without hypotheses) as an equation between urterms (without focus) on the right of a quantified implication.

Definition 11.5 The denotation of an urpredicate is an $\mathcal{M}$-map from a general $\mathcal{Q}$-object into an $\mathcal{A}$-object, defined by structural recursion as follows:
(a) The true predicate, $\top$, defines the total subspace, $[\![\vec{x} : \vec{A}, \top]\!] \cong \Pi\vec{A}$, and satisfies Axiom 9.8.
(b) When $\mathfrak{p}(\vec{x}\,)$ is an equation $\vec{x} : \vec{A} \vdash b = c : B$, the denotation is given by the equaliser,

$$[\![\vec{x} : \vec{A},\ b = c]\!] \rightarrowtail \Pi\vec{A} \underset{[\![c]\!]}{\overset{[\![b]\!]}{\rightrightarrows}} B.$$

Notice that this interprets equality at arbitrary urtypes, not just Σ, so we will have to show (in Lemma 11.12) that this agrees with Definition 9.13.

(c) Conjunction, &, is given by the intersection or pullback,

$$\begin{tikzcd}
[\![\vec{x}:\vec{A},\ \mathfrak{p}(\vec{x}\,)\ \&\ \mathfrak{q}(\vec{x}\,)]\!] \arrow[r, tail] \arrow[d, tail] \arrow[dr, phantom, "\lrcorner", very near start] & [\![\vec{x}:\vec{A},\ \mathfrak{q}(\vec{x}\,)]\!] \arrow[d, tail] \\
[\![\vec{x}:\vec{A},\ \mathfrak{p}(\vec{x}\,)]\!] \arrow[r, tail] & \Pi\vec{A}.
\end{tikzcd}$$

(d) We shall interpret $\forall\!\Rightarrow$ in Proposition 11.9.

A context $\Gamma \equiv [\vec{x}:A, \vec{\mathfrak{p}}(\vec{x}\,)]$ containing predicates is also interpreted by a subspace of the product $\prod \vec{A}$, using similar pullback diagrams.

Definition 11.6 An equideductive judgement $\Gamma \equiv [\vec{x}:\vec{A}, \vec{\mathfrak{p}}(\vec{x}\,)] \vdash \mathfrak{r}(\vec{x}\,)$ is ***valid*** in the interpretation if there is an inclusion (commutative triangle)

$$\begin{tikzcd}
[\![\Gamma]\!] \arrow[rr, tail, dotted] \arrow[dr, tail, "{[\![\mathfrak{p}]\!]}"'] & & [\![\mathfrak{r}]\!] \arrow[dl, tail, "{[\![\mathfrak{r}]\!]}"] \\
& \prod A_i &
\end{tikzcd}$$

of the subspace of $\prod \vec{A}$ that is the denotation of Γ in the denotation of $\mathfrak{r}$. The existence or otherwise of such a map is a question of fact in $\mathcal{Q}$ and is not additional structure.

We say that an equideductive category $\mathcal{Q}$ is a ***model*** of a system of *particular axioms* (Remark 9.18) if they are valid this sense. We also have to show that each of the rules is ***sound***, *i.e.* that if its premises are valid then so is its conclusion.

Lemma 11.7 An equality judgement $\Gamma \vdash b = c : B$ betweem urterms is valid in $\mathcal{Q}$ iff they have the same denotation $[\![b]\!] = [\![c]\!] : [\![\Gamma]\!] \rightrightarrows [\![B]\!]$ as morphisms of $\mathcal{Q}$.

$$\begin{tikzcd}
[\![\Gamma]\!] \arrow[d, dotted] \arrow[dr, tail] & & \\
[\![b=c]\!] \arrow[r, tail] & \prod [\![\vec{A}]\!] \arrow[r, shift left, "{[\![b]\!]}"] \arrow[r, shift right, "{[\![c]\!]}"'] & [\![B]\!]
\end{tikzcd}$$

Proof The denotations of the urterms are the composites $[\![\Gamma]\!] \rightrightarrows [\![B]\!]$ and that of the equality predicate is the equaliser. The two urterms therefore have the same denotation (these composites are the same) iff they factor through the equaliser (the predicate is valid). $\square$

Lemma 11.8 The structural rules except cut (Axiom 9.7) and the logical rules for $\top$, & and $\times$ (Axioms 9.8 and 9.16) are sound.

Proof From the diagrams in Definition 11.5,

(a) the hypothesis rule is interpreted by the inclusion from the intersection;

(b) weakening is given by pre-composition of an inclusion;

(c) reflexivity is given by the diagonal;

(d) exchange, contraction, symmetry and transitivity are properties of this intersection; and

(e) the rules for $\top$ and & are also interpreted in Definition 11.5.

This leaves products. We deduce the equideductive $\times\beta$-rule by applying Lemma 11.7 to the rule of the same name in the restricted λ-calculus, which is valid by Lemma 11.3. The $\times$-extensionality rule illustrates the interpretation of urpredicates: The denotation of the left hand side is the intersection of the equalisers

$$\bullet \rightarrowtail \Gamma \underset{q}{\overset{p}{\rightrightarrows}} A \times B \xrightarrow{\pi_0} A \quad \text{and} \quad \bullet \rightarrowtail \Gamma \underset{q}{\overset{p}{\rightrightarrows}} A \times B \xrightarrow{\pi_1} B,$$

whilst the right hand side is the equaliser $\bullet \rightarrowtail \Gamma \rightrightarrows A \times B$ of p and q, but these are isomorphic finite limits in the category $\mathcal{Q}$. $\square$

Now we add the ideas of Section 2 to the finite limits that we have used so far.

Proposition 11.9 The connectives $\forall$ and $\Rightarrow$ (Axiom 9.9)

$$\frac{\Gamma,\ y : B,\ \mathfrak{q}(y) \quad \vdash \quad \alpha xy = \beta xy}{\Gamma \quad \vdash \quad \forall y{:}B.\ \mathfrak{q}(y) \Rightarrow \alpha xy = \beta xy}$$

are soundly interpreted by the following partial product diagram (*cf.* Definition 2.1):

$$\begin{tikzcd}
 & & E \equiv [\![x : A,\ \forall y{:}B.\ \mathfrak{q}(y) \Longrightarrow \alpha xy = \beta xy]\!] \arrow[drr, "\text{bottom right}"] & & \\
{[\![\Gamma]\!]} \arrow[urr, dotted, "\text{bottom line}"] \arrow[rrrr, "{[\![a]\!]}"] & & & & {[\![A]\!]} \\
 & & E \times Y \arrow[uu] \arrow[drr] \arrow[dd] & & \\
{[\![\Gamma]\!] \times Y} \arrow[uu] \arrow[urr, dotted] \arrow[rrrr, "{[\![a]\!] \times Y}"] \arrow[drr, dotted, "\text{top line}"'] & & & & {[\![A]\!] \times Y} \arrow[uu, "Y \equiv [\![y : B,\ \mathfrak{q}(y)]\!]"'] \arrow[d, tail] \\
 & & D \equiv [\![x : A, y : B,\ \alpha xy = \beta xy]\!] \arrow[rr, tail, "\text{top right}"] & & {[\![A]\!] \times [\![B]\!]} \arrow[r, shift left, "\alpha"] \arrow[r, shift right, "\beta"'] & \Sigma
\end{tikzcd}$$

Proof We construct the partial product and show that the rules are sound. The inclusions in the middle column are contexts with urpredicates (Definition 11.5). The trapezium commutes because the composites from $E \times Y$ to Σ are the same, whilst the object D is the equaliser of α and β. Somehow this gives the $\forall E$ rule. The univeral property provides the dotted maps on the left, which asset validity of the judgements that form the $\forall I$ rule (Proposition 11.8). $\square$

The next stage brings in the categorical ideas from Section 4 (exponentials of urspaces) and the symbolic ones in Section 5 (the restricted λ-calculus), replacing parts (k,l,q) of Proposition 6.1. These depend on $\forall\Rightarrow$ because of the way in which we axiomatised the λ-calculus in Section 9.

Proposition 11.10 The rules for the λ-calculus (Axiom 9.12) are valid in this interpretation.

Proof By Lemma 11.3, urterms that are interchangeable using the $\lambda\beta$-rule have the same denotation, so the equality predicate between them is $\top$ by Lemma 11.7. By Lemma 4.10, the (*) property in Lemma 9.14,

$$\forall x.\ \phi x = \psi x \quad \dashv\vdash \quad \phi = \psi,$$

is valid in the equideductive category. Extensionality, the η- and equality-transmitting rules follow from this by Lemma 11.7 since Definition 11.5(b) interpreted equality at general urtypes, not just Σ. $\square$

Apart from Leibnizian equality and cut, this completes the interpretation of the part of the logic that we introduced in Section 9, so now we add sobriety from Sections 8 and 10.

Lemma 11.11 The operation focus of equideductive logic (Axiom 10.1 and the extensions in Lemmas **??** and 8.7) is interpreted in an equideductive category in the same way as that of the sober λ-calculus.

Proof The interpretation $[\![A]\!]$ of any urtype must be exponentiable and therefore sober, so it defines an equaliser diagram as in Theorem 8.14. The difference is that primality may now depend on equational hypotheses, because the introduction rule for focus in equideductive logic allows a general context Γ on the left. The interpretation $[\![\Gamma]\!]$ of this context may therefore be an arbitrary object of $\mathcal{Q}$, not just of $\mathcal{A}$, but in the definition of an equideductive category we required that all objects respect the universal properties. The mediator to the equaliser therefore respects the $\mathsf{focus}\,\beta$-rule (Axiom 10.1) and the other rules follow from Lemma 10.3. $\square$

Since Leibnizian equality (Definition 9.13) may be seen as the equality-transmitting rule for focus, these ideas also yield

Lemma 11.12 Equality at general urtypes is valid.

Proof One direction follows from Lemma 11.7. Conversely, let $\Gamma \vdash a, b : A$ be urterms for which the equideductive predicate $\Gamma \vdash \forall \phi.\ \phi a = \phi b$ denotes $\top$. Then $\Gamma, \phi : \Sigma^A \vdash a = b$ also denotes $\top$ by Proposition 11.9 and $\Gamma \vdash \lambda\phi.\ \phi a = \lambda\phi.\ \phi b$ denotes $\top$ by Proposition 11.10. By Lemma 11.7, the equaliser of the composites $[\![\Gamma]\!] \rightrightarrows [\![\Sigma^{\Sigma^A}]\!]$ in

$$[\![\lambda\phi.\ \phi a = \lambda\phi.\ \phi b]\!] \rightarrowtail [\![\Gamma]\!] \underset{[\![b]\!]}{\overset{[\![a]\!]}{\rightrightarrows}} [\![A]\!] \overset{\eta}{\rightarrowtail} [\![\Sigma^{\Sigma^A}]\!]$$

is $[\![\Gamma]\!]$, so these composites are the same. The interpretation $[\![A]\!]$ of a urtype in an equideductive category is an urspace, which is sober by Proposition 7.7. Therefore the map η is mono, so $[\![a]\!]$ and $[\![b]\!] : [\![\Gamma]\!] \rightrightarrows [\![A]\!]$ are also the same. Hence their equaliser is also the whole of $[\![\Gamma]\!]$, which means that the denotation of $\Gamma \vdash a = b$ is $\top$. $\square$

12 The classifying category

Now we shall construct the category of contexts and substitutions $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ for equideductive logic, as we did for the restricted λ-calculus in Definition 6.3. Then the interpretations of $\mathcal{L}$ in an equideductive category $\mathcal{Q}$ that we considered in the previous section correspond to structure-preserving functors $[\![-]\!] : \mathsf{Cn}^{\forall}_{\mathcal{L}} \to \mathcal{Q}$.

Remark 12.1 Following the example of $\mathsf{Cn}^{\lambda}_{\mathcal{L}}$ directly, we might expect the objects of $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ to be contexts and the morphisms to be strings of provable judgements like

$$\vec{x} : \vec{A},\quad \mathfrak{p}(\vec{x}) \quad\vdash\quad \vec{b} : \vec{B},\quad \mathfrak{r}(\vec{b}).$$

However, there are problems of existence and uniqueness if we do this:

(a) there need be no valid derivation of the terms $\vec{x} : \vec{A} \vdash \vec{b} : \vec{B}$ without using the hypothesis $\mathfrak{p}(\vec{x}\,)$, since formation of $\vec{b}$ may involve sub-terms $\mathsf{focus}\,P$ for which the proofs of the primality equations for the P depend on the $\mathfrak{p}(\vec{x}\,)$; and

(b) there may be many intrinsically different terms that represent what should be a single morphism.

Since the unconditional urterm $\vec{x} : \vec{A} \vdash \vec{b} : \vec{B}$ would fill in the dotted map in the square

$$\begin{tikzcd}
{[\vec{x} : \vec{A}, \mathfrak{p}(\vec{x}\,)]} \arrow[r] \arrow[d] & {[\vec{y} : \vec{B}, \mathfrak{q}(\vec{y}\,)]} \arrow[d] \\
{[\vec{x} : \vec{A}]} \arrow[r, dotted, "?"] & {[\vec{y} : \vec{B}]}
\end{tikzcd}$$

the property that we need for (a) is *injectivity* of $\vec{B}$. We studied this semantically in Section 4: the injective urspaces in **Sob** are the continuous lattices, whereas the most general ones are locally compact spaces. Definition 10.7 provided the analogous syntactic idea, which is what we shall use.

Unfortunately, we therby lose the *verbatim* interpretation of the base types of the language $\mathcal{L}$, but we shall repair this in Section 14. For (b), a morphism must be an equivalence class of urterms.

Definition 12.2 In $\mathsf{Cn}^{\forall}_{\mathcal{L}}$, *cf.* Definition 6.3 for $\mathsf{Cn}^{\lambda}_{\mathcal{L}}$,

(a) an *object* is a context

$$\big[x_1 : A_1,\ \ldots,\ x_n : A_n,\ \mathfrak{p}(\vec{x}\,)\big]$$

where $\vec{A}$ are syntactically injective urtypes and $\mathfrak{p}(\vec{x}\,)$ is an urpredicate (without focus);

(b) an *urspace* is an object for which $\mathfrak{p}(\vec{x}\,)$ is $\top$;

(c) a *morphism*

$$\vec{f} : [\vec{x} : \vec{A},\ \mathfrak{p}(\vec{x}\,)] \longrightarrow [\vec{y} : \vec{B},\ \mathfrak{r}(\vec{y}\,)]$$

is an equivalence class of strings of urterms (without focus)

$$\vec{x} : \vec{A} \vdash f_j\vec{x} : B_j \qquad \text{such that} \qquad \vec{x} : \vec{A},\ \mathfrak{p}(\vec{x}\,) \vdash \mathfrak{r}(\vec{f}\vec{x}\,),$$

where $\vec{f}$ represents the same morphism as $\vec{g}$ if

$$\vec{x} : \vec{A},\ \mathfrak{p}(\vec{x}\,) \quad \vdash \quad \vec{f}\,\vec{x} = \vec{g}\,\vec{x};$$

(d) the *identity* morphism on $[\vec{x} : \vec{A}, \mathfrak{p}(\vec{x}\,)]$ is the string $\vec{x} : \vec{A} \vdash x_j \equiv \pi_j\vec{x} : A_j$;

(e) if $\vec{x} : \vec{A}, \mathfrak{p}(\vec{x}\,) \vdash \mathfrak{q}(\vec{x}\,)$ then there is a *canonical inclusion* $[\vec{x} : \vec{A}, \mathfrak{p}(\vec{x}\,)] \rightarrowtail [\vec{x} : \vec{A}, \mathfrak{q}(\vec{x}\,)]$ that is defined in the same way as the identity;

(f) *composition* is given by substitution, as it was in $\mathsf{Cn}^{\lambda}_{\mathcal{L}}$;

(g) we write $\mathbf{1} \equiv [\top]$ for the empty urcontext with the true predicate; and

(h) we also write Σ for the object $[\sigma : \Sigma, \top]$, where the constant $\star : \Sigma$ defines a morphism $\mathbf{1} \to \Sigma$.

Lemma 12.3 The structure $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ is a category with a choice of products.

Proof To the proof of Lemma 6.4 we add that the identity is well formed because $\mathfrak{r} \equiv \mathfrak{p}$ and composition respects well-formedness by Axiom 9.7(f).

Proof It is necessary to show that "$\vec{f} = \vec{g}$" is an equivalence relation, that the identity satisfies the well-formedness condition, that composition respects both of these things and that the identity and associativity axioms hold up to equivalence. $\square$

Although we cannot interpret the language $\mathcal{L}$ directly in $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ because we have only used injective urtypes, we can lift interpretations in other categories:

Lemma 12.4 Let $[\![-]\!]$ be an interpretation of the language $\mathcal{L}$ in an equideductive category $\mathcal{Q}$. Then the interpretation of the contexts that we have used as objects and of strings of terms defines a functor $[\![-]\!] : \mathsf{Cn}^{\forall}_{\mathcal{L}} \to \mathcal{Q}$.

Proof Definition 11.5 provides the interpretation of the objects (contexts) and Remark 11.1 that of morphisms (urterms). $\square$

Along with proving that $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ has the structure of an equideductive category we shall also show that the functor $[\![-]\!]$ preserves this structure.

Lemma 12.5 The category $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ has all finite products and the functor $[\![-]\!] : \mathsf{Cn}^{\forall}_{\mathcal{L}} \to \mathcal{Q}$ preserves them. The product of two urspaces is another urspace.

Proof The terminal object is $\mathbf{1}$, to which the only incoming morphism is the empty string. Binary products in $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ are defined by

$$[\vec{x} : \vec{A},\ \mathfrak{p}(\vec{x}\,)] \ \times\ [\vec{y} : \vec{B},\ \mathfrak{r}(\vec{y}\,)] \ \equiv\ [\vec{x} : \vec{A},\ \vec{y} : \vec{B},\ \mathfrak{p}(\vec{x}\,) \,\&\, \mathfrak{r}(\vec{y}\,)],$$

so in particular the product of two urspaces is another urspace since $\top \,\&\, \top \dashv\vdash \top$. Pairing and projections are given by combining and eliminating sub-strings. Since such strings of urterms may be defined in any context, ther latter respects such products. The interpretation preserves this by Lemma 11.4. $\square$

Lemma 12.6 The category $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ has equalisers and the functor $[\![-]\!] : \mathsf{Cn}^{\forall}_{\mathcal{L}} \to \mathcal{Q}$ preserves them.

$$E \ \equiv\ [\vec{x} : \vec{A},\ \mathfrak{p}(\vec{x}\,) \,\&\, \vec{f}\vec{x} = \vec{g}\vec{x}\,] \rightarrowtail X \ \equiv\ [\vec{x} : \vec{A},\ \mathfrak{p}(\vec{x}\,)] \ \underset{\vec{g}}{\overset{\vec{f}}{\rightrightarrows}}\ Y \ \equiv\ [\vec{y} : \vec{B},\ \mathfrak{r}(\vec{y}\,)]$$

Proof The construction is illustrated by the diagram. In this, $\mathfrak{r}$ is irrelevant because we may take the equality to be at type ΠB. The interpretation $\mathsf{Cn}^{\forall}_{\mathcal{L}} \to \mathcal{Q}$ preserves this structure by Definition 11.5(b). $\square$

Lemma 12.7 The full subcategory $\mathcal{A} \subset \mathsf{Cn}^{\forall}_{\mathcal{L}}$ of urspaces has exponentials $\Sigma^{(-)}$, whose universal property is respected by all objects of $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ and preserved by $[\![-]\!]$.

Proof By λ-abstraction there is a natural bijection between morphisms

$$[\vec{y} : \vec{B},\ \mathfrak{q}(\vec{y})] \ \times\ [\vec{x} : \vec{A},\ \top] \longrightarrow \Sigma \equiv [\sigma : \Sigma, \top] \quad \text{and} \quad [\vec{y} : \vec{B},\ \mathfrak{q}(\vec{y})] \longrightarrow [\phi : \Sigma^{\vec{A}},\ \top]. \qquad \square$$

Lemma 12.8 The category $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ has partial products (Definition 2.1) and their inclusions are (isomorphic to) canonical inclusions.

$$
\begin{tikzcd}
 & E \equiv [\vec{x} : \vec{A},\ \mathfrak{r}(\vec{x}\,)] \arrow[dr, "i"] & & \\
\Gamma \equiv [\vec{z} : \vec{C},\ \mathfrak{s}(\vec{z}\,)] \arrow[ur, dotted, "\vec{f}"] \arrow[rr, "\vec{f}"] & & {[\vec{x} : \vec{A},\ \mathfrak{p}(\vec{x}\,)] \equiv X} & \\
 & E \times Y \equiv [\vec{x},\ \vec{y},\ \mathfrak{r}(\vec{x}\,) \,\&\, \mathfrak{q}(\vec{y}\,)] \arrow[uu] \arrow[dr] \arrow[dd] & & \\
\Gamma \times Y \equiv [\vec{z},\ \vec{y},\ \mathfrak{s}(\vec{z}\,) \,\&\, \mathfrak{q}(\vec{y}\,)] \arrow[uu] \arrow[ur, dotted] \arrow[rr, "\vec{f} \times Y"] \arrow[d] \arrow[dr, dotted] & & {[\vec{x},\ \vec{y},\ \mathfrak{p}(\vec{x}\,) \,\&\, \mathfrak{q}(\vec{y}\,)] \equiv X \times Y} \arrow[uu] \arrow[d] & \\
{[\vec{z} : \vec{C},\ \vec{y} : \vec{B}]} & {[\vec{x},\ \vec{y},\ \alpha\vec{x}\vec{y} = \beta\vec{x}\vec{y}]} \arrow[r, tail] & {[\vec{x} : \vec{A},\ \vec{y} : \vec{B}]} \arrow[r, shift left, "\alpha"] \arrow[r, shift right, "\beta"'] & \Sigma
\end{tikzcd}
$$

where $$\mathfrak{r}(\vec{x}\,) \ \equiv\ \mathfrak{p}(\vec{x}\,)\ \&\ \forall\vec{y}.\, \mathfrak{q}(\vec{y}\,) \Longrightarrow \alpha\vec{x}\vec{y} = \beta\vec{x}\vec{y}.$$

All objects of $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ respect partial products and the interpretation $\mathsf{Cn}^{\forall}_{\mathcal{L}} \to \mathcal{Q}$ preserves them.

Proof The universal property is tested by a morphism $\Gamma \to X$ that is a string of urterms $\vec{z} : \vec{C} \ \vdash \vec{f}\vec{z} : \vec{A}$ such that

$$\vec{z} : \vec{C},\ \mathfrak{s}(\vec{z}\,)\ \vdash\ \mathfrak{p}(\vec{f}\vec{z}\,) \quad \text{and} \quad \vec{z} : \vec{C},\ \vec{y} : \vec{B},\ \mathfrak{s}(\vec{z}\,),\ \mathfrak{q}(\vec{y}\,)\ \vdash\ \alpha(\vec{f}\vec{z}\,)\vec{y}\ =\ \beta(\vec{f}\vec{z}\,)\vec{y},$$

so $$\vec{z} : \vec{C},\ \mathfrak{s}(\vec{z}\,)\ \vdash\ \forall\vec{y}.\, \mathfrak{q}(\vec{y}\,) \Longrightarrow \alpha(\vec{f}\vec{z}\,)\vec{y}\ =\ \beta(\vec{f}\vec{z}\,)\vec{y},$$

which, together with $\mathfrak{p}(\vec{f}\vec{z}\,)$, is $\mathfrak{r}(\vec{f}\vec{z}\,)$. Hence the mediator $\Gamma \to E$ is defined by the same string $\vec{f}$. $\square$

Lemma 12.9 All objects of $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ are generated from urspaces by pullbacks and partial products, whilst all canonical inclusions arise from partial products.

Proof We use recursion on the defining urpredicate of the object, by Remark 9.6:

(a) $\{A \mid \top\}$ is an urspace;

(b) $\{A \mid \mathfrak{p} \,\&\, \mathfrak{q}\}$ is the intersection (pullback) of $\{A \mid \mathfrak{p}\}$ and $\{A \mid \mathfrak{q}\}$ rooted at A (*cf.* Definition 11.5(c)); and

(c) $\{x : A \mid \forall y.\, \mathfrak{p}(y) \Longrightarrow \alpha xy = \beta xy\}$ is a partial product whose types involve simpler predicates. $\square$

Lemma 12.10 All urspaces in the sense of Definition 12.2(b), in particular Σ, are injective. There are enough injectives, in the well founded sense of Axiom 4.6.

$$
\begin{tikzcd}
{[x : A, \mathfrak{p}(x)]} \arrow[r, tail, "\mathsf{id}"] \arrow[d, "f"'] & {[x : A, \mathfrak{q}(x)]} \arrow[r, tail, "\mathsf{id}"] \arrow[d, dotted, "f"'] & {[x : A, \top]} \arrow[d, "f"] \\
{[y : B, \top]} \arrow[r, equal] & {[y : B, \top]} \arrow[r, equal] & {[y : B, \top]}
\end{tikzcd}
$$

Proof Partial product inclusions are the same as canonical inclusions by Proposition 12.8 and Lemma 12.9. By Definition 12.2(c), a morphism $[x : A, \mathfrak{p}(x)] \to [y : B, \top]$ is represented by some urterm $x : A \vdash fx : B$, which also represents morphisms $[x : A, \top] \to [y : B, \top]$ and $[x : A, \mathfrak{q}(x)] \to [y : B, \top]$. Since the same urterm represents all of the morphisms, they make the diagram commute. Hence $[y : B, \top]$ is injective. There are enough injectives, because by Definition 12.2(a) any object is of the form $[y : B, \mathfrak{q}(y)]$ and has a canonical or partial product inclusion into $[y : B, \top]$. This is well founded by Lemma 12.9. $\square$

Proposition 12.11 The category $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ is equideductive. Moreover, any interpretation of $\mathcal{L}$ in an equideductive category $\mathcal{Q}$ extends to a functor $[\![-]\!] : \mathsf{Cn}^{\forall}_{\mathcal{L}} \to \mathcal{Q}$ that preserves the structure and this extension is unique up to unique isomorphism. $\square$

In Section 14 we shall show how to interpret the non-injective types of the language $\mathcal{L}$ in the category $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ and hence justify calling it the classifying category for the logic.

13 Completeness

We show in this section that the equideductive *logic* that we described in Sections 9–10 is complete for the notion of equideductive *category* defined in Section 4. That is, any such category $\mathcal{Q}$ is equivalent to the classifying category $\mathsf{Cn}^{\forall}_{\mathcal{L}_\forall}$ for its proper language $\mathcal{L}_\forall$. This builds on the corresponding results for the restricted λ-calculus in Section 6.

Definition 13.1 The ***proper language*** $\mathcal{L}_\forall$ of an equideductive category $\mathcal{Q}$ consists of the proper λ-calculus $\mathcal{L}_\lambda$ of its subcategory of injective objects (Definition 6.6) together with a *particular axiom* (Remark 9.18)

$$\Gamma \quad \vdash \quad \sigma \; = \; \tau \qquad \text{whenever the maps} \qquad \begin{tikzcd} {[\![\Gamma]\!]} \arrow[r, shift left, "{[\![\sigma]\!]}"] \arrow[r, shift right, "{[\![\tau]\!]}"'] & \Sigma \end{tikzcd}$$

are the same in $\mathcal{Q}$. No *global choice* of structure is necessary to state this, just the existence of the finitely many pullbacks and partial products that are needed to define *some* interpretation of Γ, σ and τ in $\mathcal{Q}$. This definition is invariant under isomorphism and therefore remains valid for other choices of pullbacks and partial products, so the maps $[\![\sigma]\!]', [\![\tau]\!]' : [\![\Gamma]\!]' \rightrightarrows \Sigma$ are also the same.

Lemma 13.2 If $\mathcal{Q}$ does have a choice of the structure for an equideductive category then there is a diagram of functors

$$\begin{tikzcd} \mathcal{A} \arrow[r, shift left, "\ulcorner\_\urcorner"] \arrow[d] & \mathsf{Cn}^{\lambda}_{\mathcal{L}_\lambda} \arrow[l, shift left, "{[\![-]\!]_\lambda}"] \arrow[l, phantom, "\simeq"] \arrow[d] \\ \mathcal{Q} & \mathsf{Cn}^{\forall}_{\mathcal{L}_\forall} \arrow[l, "{[\![-]\!]_\forall}"'] \end{tikzcd}$$

that commutes "on the nose" from $\mathcal{A}$ or $\mathsf{Cn}^{\lambda}_{\mathcal{L}_\lambda}$ to $\mathcal{Q}$, where
(a) $\mathcal{L}_\lambda$ is the proper language of $\mathcal{A}$ in the resticted λ-calculus (Definition 6.6), and
(b) $\mathsf{Cn}^{\lambda}_{\mathcal{L}_\lambda}$ is its classifying category (Definition 6.3), so
(c) $\mathcal{A} \simeq \mathsf{Cn}^{\lambda}_{\mathcal{L}_\lambda}$ with $[\![\ulcorner A \urcorner]\!] = A$ and $\eta_\Gamma : \Gamma \cong \ulcorner[\![\Gamma]\!]\urcorner$ by Theorem 6.8,
(d) $\mathcal{L}_\forall$ is the proper language of $\mathcal{Q}$ as defined above,
(e) $\mathsf{Cn}^{\forall}_{\mathcal{L}_\forall}$ is its classifying category, defined in the previous section, and
(f) $[\![-]\!]_\forall : \mathsf{Cn}^{\forall}_{\mathcal{L}_\forall} \to \mathcal{Q}$ is its interpretation, as in Section 11, since $\mathcal{Q}$ is a model of its own particular axioms. $\square$

Notation 13.3 For any context $\Gamma \equiv [\vec{z} : \vec{C}, \mathfrak{r}(\vec{z})]$, we write $\Gamma_0 \equiv [\vec{z} : \vec{C}\,]$ for the ambient *ur*context (without the predicate) and $i_\Gamma : \Gamma \rightarrowtail \Gamma_0$ for its canonical inclusion. Then $[\![\Gamma_0]\!]$ is injective in $\mathcal{Q}$ by Lemma 11.2 and $[\![i_\Gamma]\!]$ is an $\mathcal{M}$-map.

Lemma 13.4 Let Γ be any context and A any syntactically injective urtype of $\mathcal{L}_\forall$. Then any map $e : [\![\Gamma]\!] \to [\![A]\!]$ between their interpretations in $\mathcal{Q}$ is the interpretation of some urterm $\Gamma \vdash a : A$.

$$\begin{tikzcd}
{[\![\Gamma]\!]} \arrow[dd, rightarrowtail, "{[\![i_\Gamma]\!]}"'] \arrow[dr, "e"] & \\
& {[\![A]\!]} \\
{[\![\Gamma_0]\!]} \arrow[ur, dotted, "f"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
\Gamma \equiv [\vec{z} : \vec{C},\ \mathfrak{r}(\vec{z})] \arrow[dd, rightarrowtail, "i_\Gamma"'] \arrow[dr, dotted, "a"] & \\
& A \\
\Gamma_0 \equiv [\vec{z} : \vec{C}] \arrow[ur, dotted, "\ulcorner f \urcorner"'] &
\end{tikzcd}$$

Proof In the category $\mathcal{Q}$, the objects $[\![\Gamma_0]\!]$ and $[\![A]\!]$ are injective and $[\![i_\Gamma]\!] : [\![\Gamma]\!] \rightarrowtail [\![\Gamma_0]\!]$ is an $\mathcal{M}$-map, so e lifts to $f : [\![\Gamma_0]\!] \to [\![A]\!]$. Since this lies in the full subcategory $\mathcal{A} \subset \mathcal{Q}$, it is the interpretation of the urterm $\Gamma_0 \vdash a \equiv \ulcorner f \urcorner \vec{z} : A$, by Corollary 6.9. Hence the given map e is the interpretation of the weakening of a by the predicate $\mathfrak{r}(\vec{z})$. $\square$

Lemma 13.5 If the urterms $\Gamma \vdash a, b : A$ have the same interpretation as maps in $\mathcal{Q}$ then $\Gamma \vdash a = b : A$ is provable from $\mathcal{L}_\forall$.

Proof The denotations in $\mathcal{Q}$ are the composites

$$\begin{tikzcd}
{[\![\Gamma]\!]} \arrow[r, rightarrowtail, "{[\![i_\Gamma]\!]}"] & {[\![\Gamma_0]\!]} \arrow[r, shift left, "{[\![a]\!]}"] \arrow[r, shift right, "{[\![b]\!]}"'] & {[\![A]\!]}
\end{tikzcd}$$

so the following composites are also the same in $\mathcal{Q}$:

$$\begin{tikzcd}
{[\![\phi : \Sigma^A,\ \Gamma]\!]} \arrow[r, equal] & \Sigma^{[\![A]\!]} \times [\![\Gamma]\!] \arrow[r, rightarrowtail] & \Sigma^{[\![A]\!]} \times [\![\Gamma_0]\!] \arrow[r, shift left, "\mathsf{id} \times {[\![a]\!]}"] \arrow[r, shift right, "\mathsf{id} \times {[\![b]\!]}"'] & \Sigma^{[\![A]\!]} \times [\![A]\!] \arrow[r, "\mathsf{ev}"] & \Sigma.
\end{tikzcd}$$

Hence by Definition 13.1 the judgement

$$\phi : \Sigma^A, \quad \Gamma \quad \vdash \quad \phi a \ = \ \phi b$$

is an axiom of the proper language $\mathcal{L}_\forall$. The Leibnizian equality $\Gamma \vdash a = b : A$ follows from this. $\square$

Corollary 13.6 The interpretation

$$[\![-]\!] : \ \mathsf{Cn}^\forall_{\mathcal{L}_\forall}\big(\,\Gamma,\ [x : A]\,\big) \ \longrightarrow \ \mathcal{Q}\big(\ [\![\Gamma]\!],\ [\![A]\!]\ \big)$$

is full and faithful for any context Γ and syntactically injective urtype A in $\mathcal{L}_\forall$.

Proof Recall from Definition 12.2 that a typical morphism of $\mathsf{Cn}^\forall_{\mathcal{L}_\forall}$,

$$\vec{a} : [\vec{z} : \vec{C},\ \mathfrak{r}(\vec{z})] \longrightarrow [x : A],$$

is an equivalence class of urterms $\vec{z} : \vec{C} \ \vdash \ a : A$ where a represents the same morphism of $\mathsf{Cn}^\forall_{\mathcal{L}_\forall}$ as b if

$$\vec{z} : \vec{C},\ \mathfrak{r}(\vec{z}) \quad \vdash \quad a = b.$$

The result follows using the previous two lemmas. $\square$

Next we extend this result to target contexts whose predicate is a quantified implication:

Lemma 13.7 Any $\mathcal{Q}$-map $e : \llbracket \Gamma \rrbracket \to \llbracket \vec{x} : \vec{A}, \mathfrak{p}(\vec{x}\,) \rrbracket$, where

$$\mathfrak{p}(\vec{x}\,) \;\equiv\; \forall \vec{y}{:}\vec{B}.\ \mathfrak{q}(\vec{y}) \Longrightarrow \alpha\,\vec{x}\,\vec{y} \;=\; \beta\,\vec{x}\,\vec{y},$$

is the interpretation of a unique morphism of $\mathsf{Cn}^{\forall}_{\mathcal{L}_\forall}$.

$$\begin{tikzcd}
& E \equiv \llbracket \vec{x} : \vec{A}, \mathfrak{p}(\vec{x}\,) \rrbracket \arrow[dr, "\llbracket i_E \rrbracket"] & & \\
\llbracket \Gamma \rrbracket \arrow[ur, "e \in \mathcal{Q}"] \arrow[rr, "\llbracket \vec{a} \rrbracket"] & & \llbracket \vec{A} \rrbracket & \\
& E \times Y \arrow[uu] \arrow[dr] \arrow[drr, bend left] & & \\
\llbracket \Gamma \rrbracket \times Y \arrow[uu] \arrow[ur, "e \times Y"] \arrow[rr, "\llbracket \vec{a} \rrbracket \times Y"] & & \llbracket \vec{A} \rrbracket \times Y \arrow[uu] \arrow[r, shift left, "\llbracket \alpha \rrbracket"] \arrow[r, shift right, "\llbracket \beta \rrbracket"'] & \Sigma
\end{tikzcd}$$

Proof The interpretation $E \equiv \llbracket \vec{x} : \vec{A}, \mathfrak{p}(\vec{x}\,) \rrbracket$ is defined by a partial product in $\mathcal{Q}$ as shown and the map e is the mediator $\llbracket \Gamma \rrbracket \to E$. By Lemma 13.4, $\llbracket i_E \rrbracket \cdot e$ is $\llbracket \vec{a} \rrbracket$ for some string of urterms $\Gamma \vdash \vec{a} : \vec{A}$. Then, considered as a morphism of $\mathsf{Cn}^{\forall}_{\mathcal{L}_\forall}$, this string is unique by Lemma 13.5. The question is therefore whether it satisfies $\Gamma \vdash \mathfrak{p}(\vec{a}\,)$.

Since $e : \llbracket \Gamma \rrbracket \to E$ is given in $\mathcal{Q}$, the composites

$$\llbracket \Gamma,\ \vec{y} : \vec{B},\ \mathfrak{q}(\vec{y}) \rrbracket = \llbracket \Gamma \rrbracket \times Y \xrightarrow{\llbracket \vec{a} \rrbracket \times Y} \llbracket \vec{A} \rrbracket \times Y \overset{\llbracket \alpha \rrbracket}{\underset{\llbracket \beta \rrbracket}{\rightrightarrows}} \Sigma$$

are the same in $\mathcal{Q}$. Therefore, by Definition 13.1, the judgement

$$\Gamma,\ \vec{y} : \vec{B},\ \mathfrak{q}(\vec{y}) \quad \vdash \quad \alpha\,\vec{a}\,\vec{y} \;=\; \beta\,\vec{a}\,\vec{y}$$

is an axiom of the proper language $\mathcal{L}_\forall$ of $\mathcal{Q}$ and we deduce $\Gamma \vdash \mathfrak{p}(\vec{a}\,)$ by $\forall I$. $\square$

Proposition 13.8 The interpretation $\llbracket - \rrbracket : \mathsf{Cn}^{\forall}_{\mathcal{L}_\forall} \to \mathcal{Q}$ is full and faithful.

Proof It remains to show that any $\mathcal{Q}$-map $e : \llbracket \Gamma \rrbracket \to \llbracket \vec{x} : \vec{A}, \mathfrak{p}(\vec{x}\,) \rrbracket$, where $\mathfrak{p}(\vec{x}\,)$ is a *general* equideductive predicate, is the interpretation of some urterm $\Gamma \vdash a : A$ that satisfies $\Gamma \vdash \mathfrak{p}(a)$.

Proof By Remark 9.6, $\mathfrak{p}(\vec{x}\,) \dashv\vdash \mathfrak{p}_1(\vec{x}\,) \& \cdots \& \mathfrak{p}_n(\vec{x}\,)$, where each $\mathfrak{p}_k(\vec{x}\,)$ is a quantified implication. Let $i_\Delta : [\vec{x} : \vec{A}, \mathfrak{p}(\vec{x}\,)] \rightarrowtail [\vec{x} : \vec{A}]$, so $\llbracket i_\Delta \rrbracket \cdot e$ is $\llbracket \vec{a} \rrbracket$ for some unique morphism $\Gamma \to [\vec{x} : \vec{A}]$. By the previous lemma, $\Gamma \vdash \mathfrak{p}_k(\vec{a}\,)$ for each k. Alternatively, $\llbracket \vec{x} : \vec{A}, \mathfrak{p}(\vec{x}\,) \rrbracket$ is the pullback of the $\llbracket \vec{x} : \vec{A}, \mathfrak{p}_k(\vec{x}\,) \rrbracket$ rooted at $\llbracket \vec{x} : \vec{A} \rrbracket$. $\square$

Theorem 13.9 Any equideductive category $\mathcal{Q}$ with a choice of structure is strongly equivalent to the classifying category for its proper language.

Proof For weak equivalence it remains to observe that $\llbracket - \rrbracket$ is essentially surjective (each object of $\mathcal{Q}$ is isomorphic to the interpretation of some context in $\mathcal{L}_\forall$), because this is the *well founded* version of the requirement that there be enough injectives in an equideductive category. If we strengthen this requirement to make a *choice* of injectives then this choice provides the object part of the pseudo-inverse functor. □

Theorem 13.10 Let $\mathcal{Q}$ be an equideductive category without the requirement of a choice of structure and $\mathcal{L}_\forall$ its proper language as defined above. Then $\mathcal{Q}$ is equivalent to $\mathsf{Cn}^\forall_{\mathcal{L}_\forall}$ in the very weak sense that there is a span of functors $\mathsf{Cn}^\forall_{\mathcal{L}_\forall} \leftarrow \mathcal{P} \to \mathcal{Q}$, each of which is full, faithful and essentially surjective.

Proof An object of $\mathcal{P}$ is a context Γ (object of $\mathsf{Cn}^\forall_{\mathcal{L}_\forall}$) together with a diagram in $\mathcal{Q}$. That is, an assignment of objects and morphisms of $\mathcal{Q}$ that are interpretations of the types and terms in the process of interpreting Γ in $\mathcal{Q}$. For this, $\mathcal{Q}$ does not need a *global choice* of this structure: each finite assignment that has the relevant universal properties provides one of the objects of $\mathcal{P}$. The object parts of the functors $\mathsf{Cn}^\forall_{\mathcal{L}_\forall} \leftarrow \mathcal{P} \to \mathcal{Q}$ select the whole context Γ and its interpretation in $\mathcal{Q}$. The morphisms between objects of $\mathcal{P}$ must agree with both those in $\mathsf{Cn}^\forall_{\mathcal{L}_\forall}$ and in $\mathcal{Q}$. This is valid by a finite fragment of the proof of Proposition 13.8. □

14 Comprehension types

Whilst injectives considerably simplify foundational constructions, for mathematical applications we would like to be able to use non-injective urtypes and even treat the two-part contexts (urtypes with predicates) as first class objects. In this section we introduce a notation like subset-formation or comprehension in set theory and show how to define morphisms. For simplicity and for the same reasons that underly the adoption of the variable-binding rule (Warning **??**), we do not allow dependent types here.

Definition 14.1 Let A be an urtype and $\mathfrak{p}$ a predicate on it. Then the ***type*** $\{x : A \mid \mathfrak{p}(x)\}$, or $\{A \mid \mathfrak{p}\}$ for short, is formed by the rule

$$\frac{A\ \mathsf{type} \qquad \mathfrak{p}\ \text{predicate on}\ A}{\{x : A \mid \mathfrak{p}(x)\}\ \mathsf{type}} \qquad \{\}F$$

Terms of this type obey the introduction rule

$$\frac{\Gamma\ \vdash\ a : A \qquad \Gamma\ \vdash\ \mathfrak{p}(a)}{\Gamma\ \vdash\ \mathsf{admit}\, a : \{A \mid \mathfrak{p}\}} \qquad \{\}I$$

the elimination rules

$$x : \{A \mid \mathfrak{p}\}\ \vdash\ ix : A, \qquad \mathfrak{p}(ix) \qquad \{\}E$$

and the β-rule

$$x : A, \quad \mathfrak{p}(x) \quad \vdash \quad i(\mathsf{admit}\, x)\ =\ x. \qquad \{\}\beta$$

The equality predicate for terms of type $\{A \mid \mathfrak{p}\}$ is defined in the same way as in Definition 9.13:

$$a = b : \{A \mid \mathfrak{p}\} \quad \equiv \quad \forall \phi{:}\Sigma^A.\ \phi(ia) = \psi(ia).$$

These new *types* are interpreted in an equideductive category in the same the same way as are contexts (Definition 11.5). In the topological model they are therefore general sober spaces, whereas the *urtypes* denote locally compact spaces.

Lemma 14.2 Equality is reflexive, symmetric and transitive. It obeys extensional and η-rules and it is transmitted by the introduction and elimination rules.

Proof The $\{\}E{=}$ rule follows from the definition of equality,

$$a = b : \{A \mid \mathfrak{p}\} \quad \vdash \quad ia = ib : A,$$

as do reflexivity, symmetry and transitivity. The $\{\}I{=}$ and η-rules

$$a = b : A \ \vdash \ \mathsf{admit}\, a = \mathsf{admit}\, b : \{A \mid \mathfrak{p}\} \quad \text{and} \quad a \ = \ \mathsf{admit}(ix) : \{A \mid \mathfrak{p}\}$$

follow from this and the β-rule. The extensionality rule is

$$ia = ib : A \quad \vdash \quad a = b : \{A \mid \mathfrak{p}\}. \qquad \square$$

Definition 14.3 The subtyping notation may be nested. For example,

$$\begin{array}{rl} & \{x : \{\alpha : \Sigma^A, \beta : \Sigma^B \mid \mathfrak{p}(\alpha, \beta)\},\ y : \{\gamma : \Sigma^C, \delta : \Sigma^D \mid \mathfrak{q}(\gamma, \delta)\} \mid \mathfrak{r}(x, y)\} \\ \equiv & \{\alpha : \Sigma^A, \beta : \Sigma^B, \gamma : \Sigma^C, \delta : \Sigma^D \mid \mathfrak{p}(\alpha, \beta) \,\&\, \mathfrak{q}(\gamma, \delta) \,\&\, \mathfrak{r}(\langle \alpha, \beta\rangle, \langle \gamma, \delta\rangle)\}. \end{array}$$

This is justified by showing that the nested introduction, elimination and equality rules agree with the composite ones.

Proposition 14.4 The range of the quantifier may be a type:

$$\forall y{:}\{B \mid \mathfrak{p}\}.\ \mathfrak{q}(y) \Rightarrow \mathfrak{r}(x, y) \quad \equiv \quad \forall y{:}B.\ \mathfrak{p}(y) \,\&\, \mathfrak{q}(y) \Rightarrow \mathfrak{r}(x, y).$$

Indeed, it may be a clearer to think of quantified implication as quantification without implication but over a type.

Proof The following judgements are equivalent,

$$\begin{array}{lcl} x : A,\ \mathfrak{s}(x),\ y : \{B \mid \mathfrak{p}\},\ \mathfrak{q}(y) & \vdash & \mathfrak{r}(x, y) \\ x : A,\ y : B, \quad \mathfrak{s}(x),\ \mathfrak{p}(y),\ \mathfrak{q}(y) & \vdash & \mathfrak{r}(x, y) \\ x : A,\ \mathfrak{s}(x) & \vdash & \forall y.\ \mathfrak{p}(y) \,\&\, \mathfrak{q}(y) \Longrightarrow \mathfrak{r}(x, y), \end{array}$$

by Definition 14.1 and Proposition 9.11, so the extended $\forall$ rules are satisfied. $\square$

Proposition 14.5 $\forall y{:}\mathbf{1}.\ \mathfrak{p}(x, y) \dashv\vdash \mathfrak{p}(x, \star)$ and $\forall yz.\ \mathfrak{p}(x, y, z) \dashv\vdash \forall y.\, \forall z.\ \mathfrak{p}(x, y, z)$.

Proof The equivalence

$$\frac{x : A,\ \mathfrak{s}(x),\ y : \mathbf{1} \quad \vdash \quad \mathfrak{p}(x, y)}{x : A,\ \mathfrak{s}(x) \quad \vdash \quad \mathfrak{p}(x, \star)}$$

downwards is given by $\forall E$ and upwards by $\mathbf{1}$-ext ($y : \mathbf{1} \vdash y = \star$) and $\forall I$. The judgements

$$\begin{array}{lcl} x : A,\ \mathfrak{s}(x) & \vdash & \forall \langle y, z\rangle{:}Y \times Z.\ \mathfrak{p}(x, y, z) \\ x : A,\ \mathfrak{s}(x),\ \langle y, z\rangle : Y \times Z & \vdash & \mathfrak{p}(x, y, z) \\ x : A,\ \mathfrak{s}(x),\ y : Y,\ z : Z & \vdash & \mathfrak{p}(x, y, z) \\ x : A,\ \mathfrak{s}(x),\ y : Y & \vdash & \forall z{:}Z.\ \mathfrak{p}(x, y, z) \\ x : A,\ \mathfrak{s}(x) & \vdash & \forall y{:}Y.\ \forall z{:}Z.\ \mathfrak{p}(x, y, z) \end{array}$$

are also equivalent by the previous result and the rules for pairing. $\square$

Now we would like to treat types as objects of a category, for which we need to define the morphisms. However, now that we have dropped the injectivity assumption, we must face up to the difficulty that we mentioned in Remark 12.1.

Lemma 14.6 For any (not necessarily injective) urtype A and predicate $\mathfrak{p}$ on it, the diagram

$$[P : \Sigma^{\Sigma^A},\ \mathsf{prime}(P)\ \&\ \bar{\mathfrak{p}}(P)] \rightarrowtail [P : \Sigma^{\Sigma^A},\ \bar{\mathfrak{p}}(P)] \rightrightarrows [\mathcal{F} : \Sigma^4 A,\ \bar{\bar{\mathfrak{p}}}(\mathcal{F})].$$

is an equaliser in $\mathsf{Cn}^{\forall}_{\mathcal{L}}$. Also, there is a natural bijection between the terms of these types:

$$\{x : A \mid \mathfrak{p}(x)\} \xrightarrow{\eta} \{P : \Sigma^{\Sigma^A} \mid \mathsf{prime}(P)\ \&\ \bar{\mathfrak{p}}(P)\}.$$

Proof Lemmas 12.6 and 10.14. $\square$

Proposition 14.7 Types provide the objects of a category $\mathcal{C}$ that is strongly equivalent to $\mathsf{Cn}^{\forall}_{\mathcal{L}}$. In this, a morphism

$$\{x : A \mid \mathfrak{p}(x)\} \longrightarrow \{y : B \mid \mathfrak{q}(y)\}$$

is an equivalence class of urterms

$$x : A \vdash Qx : \Sigma^{\Sigma^B} \qquad \text{for which} \qquad x : A,\ \mathfrak{p}(x)\ \vdash\ \mathsf{prime}(Qx)\ \&\ \bar{\mathfrak{q}}(Qx),$$

where $Q_1 x = Q_2 x$ if

$$x : A, \quad \mathfrak{p}(x) \quad \vdash \quad Q_1 x = Q_2 x : \Sigma^{\Sigma^B}.$$

The identity on $\{A \mid \mathfrak{p}\}$ has $Qx \equiv \lambda\phi.\ \phi x$ and the composite of Px with Qy is

$$x : A \quad \vdash \quad Rx\ \equiv\ \lambda\theta.\, P(\lambda y.\, Qy\theta) : \Sigma^{\Sigma^C}.$$

Proof If B is syntactically injective then by Proposition 10.10 there is (unconditionally) an urterm $b : B$ for which $Q = \lambda\psi.\ \psi b$ when $\mathsf{prime}(Q)$. Hence there is a full and faithful functor $\mathsf{Cn}^{\forall}_{\mathcal{L}} \to \mathcal{C}$. Both of these categories have equalisers, whilst by the Lemma both $[x : A, \mathfrak{p}(x)] \in \mathcal{C}$ and $[P : \Sigma^{\Sigma^X}, \bar{(}P)\ \&\ \mathsf{prime}(P)] \in \mathsf{Cn}^{\forall}_{\mathcal{L}} \subset \mathcal{C}$ are equalisers of the same pair of maps in $\mathcal{C}$.

The typical morphism of $\mathcal{C}$ must be defined like this:

$$\begin{tikzcd}
\{x : A \mid \mathfrak{p}(x)\} \arrow[rr, "{b : B,\ \mathfrak{q}(b)}"] \arrow[d, equal, "\eta_A"'] & & \{y : B \mid \mathfrak{q}(y)\} \arrow[d, equal, "\eta_B"] \\
{[P : \Sigma^{\Sigma^A},\ \bar{\mathfrak{p}}(P)\ \&\ \mathsf{prime}(P)]} \arrow[rr] & & {[Q : \Sigma^{\Sigma^B},\ \bar{\mathfrak{q}}(Q)\ \&\ \mathsf{prime}(Q)]}
\end{tikzcd}$$

By the equivalence relation on representing urterms of morphisms of $\mathsf{Cn}^{\forall}_{\mathcal{L}}$, the lower map is represented more simply by its values on $x : A$, whose characterisation is as given. The identity and composite may be verified by λ-calculations.

Hence they are isomorphic and we have an equivalence of categories. $\square$

Theorem 14.8 The ***classifying category*** for equideductive logic is $\mathsf{Cn}^{\forall}_{\mathcal{L}}$, *cf.* Theorem 6.5:

(a) $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ is itself an equideductive category;

(b) equideductive logic and the language $\mathcal{L}$ are interpreted in $\mathcal{C} \simeq \mathsf{Cn}^{\forall}_{\mathcal{L}}$;

(c) any interpretation $[\![-]\!]$ of the logic and $\mathcal{L}$ in an equideductive category $\mathcal{Q}$ extends to a functor $[\![-]\!] : \mathsf{Cn}^{\forall}_{\mathcal{L}} \to \mathcal{Q}$ that preserves this structure, uniquely up to unique isomorphism; and

(d) any such functor restricts to an interpretation of the logic, uniquely up to unique isomorphism. $\square$

Remark 14.9 The construction that we used in Lemma 14.6,

$$T\,[x : A, \mathfrak{p}(x)] \;\equiv\; [P : \Sigma^{\Sigma^A},\ \bar{\mathfrak{p}}(P)],$$

is part of an endofunctor, indeed a monad, on $\mathsf{Cn}^{\forall}_{\mathcal{L}}$ that extends $\Sigma^{\Sigma^{(-)}}$. We shall see in [?] that it actually provides the double exponential of $X \in \mathcal{Q} \subset \mathcal{S}$ in the enclosing cartesian closed category $\mathcal{S}$. This is another way in which an equideductive category "lies nicely" within its cartesian closed extension. The Lemma showed that any such $X \in \mathcal{Q} \subset \mathcal{S}$ is sober with respect to this extended exponential.

References

[CR00] Aurelio Carboni and Giuseppe Rosolini. Locally cartesian closed and exact completions. *Journal of Pure and Applied Algebra*, 154:103–116, 2000.

[DT87] Roy Dyckhoff and Walter Tholen. Exponentiable maps, partial products and pullback complements. *Journal of Pure and Applied Algebra*, 49:103–116, 1987.

[Füh99] Carsten Führmann. Direct models of the computational lambda-calculus. In *Mathematical Foundations of Programming Semantics 15*, number 20 in Electronic Notes in Theoretical Computer Science, 1999.

[Hyl91] J. Martin E. Hyland. First steps in synthetic domain theory. In Aurelio Carboni, Maria-Cristina Pedicchio, and Giuseppe Rosolini, editors, *Proceedings of the 1990 Como Category Conference*, number 1488 in Lecture Notes in Mathematics, pages 131–156. Springer-Verlag, 1991.

[Joh82] Peter T. Johnstone. *Stone Spaces*. Number 3 in Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1982.

[Kle43] Stephen Kleene. Recursive predicates and quantifiers. *Transations of the American Mathematical Society*, 53:41–73, 1943. Reprinted in *The Undecidable*, pages 254–287, edited by Davis, 1965.

[Mac71] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, Berlin, 1971.

[Nie82] Susan Niefield. Cartesianness: Topological spaces, uniform spaces and affine varieties. *Journal of Pure and Applied Algebra*, 23:147–167, 1982.

[Pas65] Boris Pasynkov. Partial topological products. *Transactions of the Moscow Mathematical Society*, 13:153–271, 1965.

[Ros86] Giuseppe Rosolini. *Continuity and Effectiveness in Topoi*. D. phil. thesis, University of Oxford, 1986.

[Sco72] Dana S. Scott. Continuous lattices. In F. W. Lawvere, editor, *Toposes, Algebraic Geometry and Logic*, number 274 in Lecture Notes in Mathematics, pages 97–136. Springer-Verlag, Berlin, 1972.

[Sel01] Peter Selinger. Control categories and duality. *Mathematical Structures in Computer Science*, 11:207–260, 2001.

[Tay91] Paul Taylor. The fixed point property in synthetic domain theory. In Gilles Kahn, editor, *Logic in Computer Science 6*, pages 152–160. IEEE, 1991.

[Tay99] Paul Taylor. *Practical Foundations of Mathematics*. Number 59 in Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1999.

[Thi97] Hayo Thielecke. *Categorical Structure of Continuation Passing Style*. PhD thesis, University of Edinburgh, 1997.

The papers on abstract Stone duality may be obtained from

`www.Paul Taylor.EU/ASD`

[O] Paul Taylor, Foundations *for* Computable Topology. in Giovanni Sommaruga (ed.), *Foundational Theories of Mathematics*, Kluwer 2011.

[A] Paul Taylor, Sober spaces and continuations. *Theory and Applications of Categories*, 10(12):248–299, 2002.

[B] Paul Taylor, Subspaces in abstract Stone duality. *Theory and Applications of Categories*, 10(13):300–366, 2002.

[C] Paul Taylor, Geometric and higher order logic using abstract Stone duality. *Theory and Applications of Categories*, 7(15):284–338, 2000.

[E] Paul Taylor, Inside every model of Abstract Stone Duality lies an Arithmetic Universe. *Electronic Notes in Theoretical Computer Science* **122** (2005) 247-296.

[G] Paul Taylor, Computably based locally compact spaces. *Logical Methods in Computer Science*, **2** (2006) 1–70.

[I] Andrej Bauer and Paul Taylor, The Dedekind reals in abstract Stone duality. *Mathematical Structures in Computer Science*, **19** (2009) 757–838.

[J] Paul Taylor, A λ-calculus for real analysis. *Journal of Logic and Analysis*, **2**(5), 1–115 (2010)

[K] Paul Taylor, Interval analysis without intervals. February 2006.

[L] Paul Taylor, Tychonov's theorem in abstract Stone duality. September 2004.

[N] Paul Taylor, Computability in locally compact spaces. 2010.

[BB] Paul Taylor, An existential quantifier for topology. 2010.

[DD] Paul Taylor, The Phoa principle in equideductive topology. 2010.