Subspaces in Abstract Stone Duality

Paul Taylor

August 11, 2003

Abstract

By *abstract* Stone duality we mean that the topology or contravariant powerset functor, seen as a self-adjoint exponential $\Sigma^{(-)}$ on some category, is monadic. Using Beck's theorem, this means that certain equalisers exist and carry the subspace topology. These subspaces are encoded by idempotents that play a role similar to that of nuclei in locale theory.

Paré showed that any elementary topos has this duality, and we prove it intuitionistically for the category of locally compact locales.

The paper is largely concerned with the construction of such a category out of one that merely has powers of some fixed object Σ. It builds on *Sober Spaces and Continuations*, where the related but weaker notion of abstract sobriety was considered. The construction is done first by formally adjoining certain equalisers that $\Sigma^{(-)}$ takes to coequalisers, then using Eilenberg–Moore algebras, and finally presented as a lambda calculus similar to the axiom of comprehension in set theory.

The comprehension calculus has a normalisation theorem, by which every type can be embedded as a subspace of a type formed without comprehension, and terms also normalise in a simple way. The symbolic and categorical structures are thereby shown to be equivalent.

Finally, sums and certain quotients are constructed using the comprehension calculus, giving an extensive category.

Contents

1 Introduction

The existence or otherwise of extensional subspaces is a point of demarcation between pure mathematics and computer science. The ubiquitous axiom of comprehension seems to be what mainstream mathematicians have in mind when they insist that Set Theory provides the foundations of their subject. On the other hand, neither programming languages nor the type theories that are most popular in computer science admit subtypes. (We do *not* mean the notion of this name in object-oriented programming, which is like an extension of the signature of an algebraic theory.) Subtypes in our sense *are* used in software development, in the form of Floyd–Hoare logic, with pre- and post-conditions, but it is an up-hill struggle to teach programmers to use them!

The reason for this demarcation is that the essence of the development in a mathematical argument is quite different from that in a computation. Before the referee will allow me to say that an element belongs to a subset, I must prove that it satisfies the defining property. Otherwise, the rest of the argument has no meaning whatever, and my paper is not published. From this point of view, compilation of high level programming languages lies on the mathematical side of the boundary. A (low level) computation, on the other hand, proceeds irrespectively of whether or not a purported mid-condition is satisfied: if it doesn't, the behaviour of the program may not be what the programmer had in mind, but it nevertheless does *something*.

I am, of course, in favour of *constructive* mathematics, and appreciate the analogy between programs and proofs, but opinions differ as to what is regarded as constructive: my notion is stronger than Johnstone's [Joh82] but weaker than Martin-Löf's [ML84]. I have argued that what Per Martin-Löf calls the "existential quantifier" and the "axiom of choice" are not those used in geometry [Tay99, Section 2.4], and nor do I consider that an element of a subset needs to be *accompanied* by a proof of the defining predicate. Such proofs may serve as guarantees, but at some point we must eventually *trust* (the verification of) the guarantee and carry on with the computation.

In crossing the demarcation from pure mathematics to programming, we therefore have to extend the definition of functions that were intended to be defined on a subset to the whole of the ambient set, possibly at the cost of results that lie outside the intended target. A target object that permits such extension without itself being expanded is called *injective*, and such objects will play an important role in this paper.

I see the crossing as building a "mathematically comfortable" *platform*, with more types (sets, spaces) than the underlying programming language, but such that the terms (elements) become programs when the richer type information is erased. The programming language that we shall use was justified and developed in the previous paper [A], so now we must consider ways of building mathematical platforms. Category theory provides the language in which to express both our requirements for "comfort" and the means of satisfying them, although equivalent λ-calculi are needed for the idioms of both mathematics and programming.

Programming languages include constructors for making new types from old ones, but the additional mathematical types depend on *terms*. A very general theory of such dependent types is formulated in [Tay99, Chapter VIII], although I now consider that to be too complicated. Simpler notions than the axiom of comprehension are familiar in category theory: equalisers, pullbacks, coequalisers and pushouts.

The present work is by no means the first to provide such extensions. One very simple idea is to split idempotents (Remark 4.1), which Dana Scott developed [Sco76]. However, a great deal of artificiality was piled on top of this, notably sum types with spurious additional elements: we shall find in Section 11 that stable disjoint coproducts are a mathematical comfort that we may reasonably demand as standard.

Another idea is to expand the *monoid* of (partial general) recursive functions on $\mathbb{N}$ to the *category* whose objects are recursively enumerable subsets. Giuseppe Rosolini formulated the abstract notion of partial maps (composing a *p-category*), and constructed a category with subobjects from it [RR88]. Peter Freyd and Andre Scedrov started instead from binary relations (composing an *allegory*) [FS90]. Their theory also accounts for the way in which general sets may be obtained from iterated powersets, traditionally known as the von Neumann hierarchy. Bill Lawvere's treatment [Law70] describes and generalises the behaviour of comprehension in toposes and other categories in which it already exists, but does not explain how it *creates* new sets [Tay99, Exercises 9.45ff]. Other ways of expanding the class of types include the regular and exact completions of categories.

The particular construction performed in this paper is based on a monad. In the original *mathematical* development of monads, Eilenberg–Moore algebras were given a preferential role [Eck69, BW85]. Later, Eugenio Moggi demonstrated how they could be used to encode features of *computation* [Mog91], but the new types that he introduced were objects of the *Kleisli* category, which consists of just the *free* algebras. So the literature that has ensued illustrates the demarcation between mathematics and computer science. In particular, the basic form of our construction without the subspaces (Section[1] A 6) is similar to work of Hayo Thielecke [Thi97], which uses the the Kleisli category.

In none of these constructions is a type defined as a "collection", as Solomon Feferman claimed to be a necessary ingredient of mathematical foundations [Fef77], and nor is a subtype a subcollection selected by some kind of demon. Both the underlying programming language and its mathematical enrichment may be defined as λ-calculi with certain syntactic rules of formation.

[1] My papers *Sober Spaces and Continuations* and *Geometric and Higher Order Logic* are cited as if they were respectively "Chapters" A and C of a book, this paper being Chapter B.

Of course, there is a (recursive) collection of well formed formulae (or of objects and morphisms in the category), but this cannot be what Feferman's argument means, as it applies equally to Set Theory itself. The metalanguage of category theory and type theory is not only first order, as is that of Set Theory, but essentially algebraic. In our case, the universal properties of Σ-split (co)equalisers are *equational*, so the quantifier complexity is even less.

Many of the mathematical intuitions on which our calculus is based come from general topology, regarded as a more subtle form of set theory. This research programme is named after Marshall Stone because he was the first to emphasise that one should always look for the *topology* on any mathematical object that one has constructed, since the appropriate morphisms are *continuous* functions, not arbitrary set-theoretic ones *à la* Cantor. (See the Introduction to [Joh82] for a historical survey of this point of view.)

Contrary to what Richard Dedekind and Georg Cantor have told us, *this is how the real line* $\mathbb{R}$ *is defined*: we understand the *topology* $\Sigma^{\mathbb{R}}$ because this is the abstraction of the convergence of sequences of rationals (or of otherwise algebraically definable numbers), and this is the *only* way by which we gain access to transcendental numbers. The space $\mathbb{R}$ exists by *fiat*, being defined *formally* as $\mathsf{pts}(\Sigma^{\mathbb{R}})$. Indeed, Bourbaki also constructed $\mathbb{R}$ at the end of a tortuous chain of definitions that spans an entire volume [Bou66], but essentially recovers it from a topology that is defined in terms of $\mathbb{Q}$.

It was Dana Scott who promoted the analogy between continuity and *computability*. However, there cannot be a precise connection with the *traditional* axiomatisation of general topology, because it relies on *arbitrary* unions, rather than recursive ones. It is principally this problem that Abstract Stone Duality seeks to address.

Monads provide a way of handling infinitary algebra of the kind that we need to re-axiomatise topology, and may be defined over any category, not just the category of sets. Specifically, we define the algebras that replace the lattices of open subsets of a space using a monad over the category of spaces itself.

This use of monads was also inspired by Marshall Stone's work. By *Stone duality* we understand the dual equivalence between a category of algebraically defined objects and another whose objects we think of as "spaces", though they may be sets, posets or algebraic varieties as well as topological spaces.

By *abstract* Stone duality we mean some category $\mathcal{C}$ of "spaces" whose opposite category $\mathcal{C}^{\mathsf{op}} \simeq \mathcal{A}$ is itself "algebraic" over $\mathcal{C}$ in the sense of being the category of Eilenberg–Moore algebras over $\mathcal{C}$.

The analogy between the two concrete examples of sets and locally compact spaces drives the intuition behind this programme. Temporarily using classical logic, including the axiom of choice, to present this intuition, these two monadic situations are illustrated by the following diagrams.

$$
\begin{tikzcd}
\mathbf{Set}^{\mathsf{op}} \arrow[r, shift left, "\text{powerset}"] \arrow[r, phantom, "\simeq"] \arrow[d, shift left, "\Omega^{(-)}"] \arrow[d, phantom, "\dashv"] & \mathbf{CABA} \arrow[l, shift left, "\text{atoms}"] \arrow[d, shift left, "U"] \arrow[d, phantom, "\dashv"] \\
\mathbf{Set} \arrow[u, shift left, "\Omega^{(-)}"] \arrow[r, equal] & \mathbf{Set} \arrow[u, shift left, dotted]
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{LKSp}^{\mathsf{op}} \arrow[r, shift left, "\text{topology}"] \arrow[r, phantom, "\simeq"] \arrow[d, shift left, "\Sigma^{(-)}"] \arrow[d, phantom, "\dashv"] & \mathbf{LKFrm} \arrow[l, shift left, "\mathsf{pts}"] \arrow[d, shift left, "U"] \arrow[d, phantom, "\dashv"] \\
\mathbf{LKSp} \arrow[u, shift left, "\Sigma^{(-)}"] \arrow[r, equal] & \mathbf{LKSp} \arrow[u, shift left, dotted]
\end{tikzcd}
$$

Adolf Lindenbaum and Alfred Tarski characterised full powerset lattices classically as complete atomic Boolean algebras [Tar35]. In modern terms, we formulate their result as a mutually inverse pair of functors, one of which takes a set X to its powerset $\mathcal{P}X$, and the other extracts the set of "atomic" elements of any complete atomic Boolean algebra.

The morphisms of **CABA** preserve arbitrary meets and joins, and the forgetful functor $U : \mathbf{CABA} \to \mathbf{Set}$ from this category has a left adjoint. The free complete atomic Boolean algebra on a set X is its double powerset, $\mathcal{P}\mathcal{P}X$, and it may be shown that this adjunction is monadic.

The equivalence of categories re-states this concrete adjunction in more abstract terms: the contravariant powerset functor $\mathcal{P} : \mathbf{Set}^{\mathsf{op}} \to \mathbf{Set}$ is *self*-adjoint, and (since the squares commute) the Lindenbaum–Tarski theorem says that this adjunction is also monadic. Robert Paré proved the same result intuitionistically for any elementary topos [Par74]. Classically, $\mathcal{P}X \cong \mathbf{2}^X$, whilst $\mathcal{P}X \cong \Omega^X$ in a topos, where Ω denotes the type of truth values.

In the second diagram, **Frm** is the category of ***frames***, *i.e.* lattices with arbitrary joins over which binary meets distribute, and homomorphisms that preserve *finite* meets and arbitrary joins. Frames are therefore the kind of lattices that open subsets constitute in general topology, whilst the homomorphisms capture the properties of inverse image maps for continuous functions. **LKSp** and **LKFrm** are the categories of locally compact sober spaces and the frames that arise as their topologies, namely those that are also continuous lattices [HM81].

LKSp and **LKFrm** are dual categories, where the "points" of a frame may be characterised lattice-theoretically [Joh82, Section II 1] or by means of an exponential $\Sigma^{(-)}$ and its associated λ-calculus [A]. **Frm** is also a category of Eilenberg–Moore algebras for a monad over **Set**. However, if U forgets just the finitary lattice structure, but leaves the carrier equipped with its directed joins and so the Scott topology, **LKFrm** is monadic over **LKSp**.

Again, the monadic adjunction may be seen abstractly, where the powerset $\mathcal{P}X \cong \Omega^X$ is replaced by the topology, *i.e.* the lattice of open subsets of X, and the lattice itself is regarded as a space with the Scott topology. The topology may be expressed as Σ^X, where Σ is the ***Sierpiński space***. Classically Σ also has two points, one of which is open.

The theorems of Stone, Tarski and Lindenbaum that we have given identified "atoms" or "points" in the lattices as sets of sets. This turns the set-theoretic membership relation on its head: each "open set" (element of the algebra) *belongs to* certain "points", *i.e.* these correspond to certain subsets of the algebra. However, the existence of *enough* ideal points to separate the elements of the algebra is equivalent to the axiom of choice, or at least to some weaker proper axiom such as the prime ideal theorem. Just the same problem arises in model theory, models being the "ideal points" of the predicate calculus.

In the following two sections we show how, when the category of "spaces" satisfies the abstract Stone duality, it admits subspaces that carry the subspace topology. This is a special case of Jon Beck's general characterisation of monadic adjunctions: under the duality, subspaces correspond to quotient algebras, and Beck's theorem says that we may calculate certain kinds of quotient algebras simply as quotients of their carriers. For textbook accounts of monads and Beck's theorem, see [ML71, Section VI 7], [BW85, Section 3.3] or [Tay99, Section 7.5]. Our discussion in Section 3 is partly intended to amplify these accounts by giving a worked example of Beck's theorem, in the hope that it will cross the demarcation and become better understood in the theoretical computer science community.

Theorem 3.11 gives an *intuitionistic* proof of Stone duality for locally compact locales. After this, there is no more topology in this paper: that story — in particular, the equivalence between Eilenberg–Moore and frame homomorphisms — is taken up in [C, D, F].

Sections 4–6 build a new category $\overline{\mathcal{C}}$ with the abstract Stone duality from one that simply has products, powers of Σ and all objects sober. The construction is based directly on Beck's theorem, formally adding Σ-split equalisers. The main difficulty is in showing that $\overline{\mathcal{C}}$ has products, for which injective objects are needed as a tool.

Section 7 briefly describes an alternative construction that uses algebras instead.

Sections 8–10 investigate an equivalent λ-calculus, whose *types* admit subspaces that are described using something like the axiom of comprehension in set theory. However, these type annotations, like the mid-conditions in Floyd–Hoare logic, are *computationally anodyne*: they remain in the journal papers and are not used at run-time. The *terms* may be stripped of this complex type information, leaving a simple λ-calculus with application and abstraction.

The final section constructs stable disjoint coproducts and Σ-split coequalisers.

Some comment is in order on the fact that locally compact locales, rather than *all* locales, provide the leading model of our axioms. Fundamentally, it is the *category* that we seek to axiomatise, not the spaces, and one category has the monadic property, whilst the other doesn't. Indeed, the axioms for locales hardly enjoy the same unanimous acceptance in the mathematical community as do those for, say, groups. *Spatial* locales agree with *sober* spaces, but even then disagree as to their products outside the locally compact case, whilst there are numerous competing axiomatisations of convergence and other features of topology. The traditional one has also long been regarded as unsatisfactory on the grounds that its *logical* properties (such as cartesian closure) are inadequate. This paper breaks new ground in treating topology as λ-calculus.

The simple technical reason for the "restriction" is that the exponential Σ^X is only defined when X is locally compact. However, Steven Vickers has shown how the functor $\Sigma^{\Sigma^{(-)}}$ may be extended to all locales, decomposing it into two covariant functors that are of interest in themselves [Vic01]. However, the Eilenberg–Moore category for his monad is not the category of frames, but one whose opposite he calls the category of *colocales*. When X is a locale, Σ^X is a colocale, and *vice versa*, so observations belong not merely to a different *type* from values, but to another *category*, whilst meta-observations are again like values. Our experience since Turing that computations are ultimately untyped seems to have been sacrificed on the altar of conformity with a pre-existing mathematical structure.

I acknowledge, on the other hand, that my mathematical platform is not as comfortable as we might like. In particular, the *intersection* of two Σ-split subspaces need not be another locally compact space. Before we are allowed to form an equaliser or coequaliser, we have to satisfy a very awkward property, which turns out to have no computational meaning anyway. (It also leads to mis-conceived questions such as largest cartesian closed categories: the statement that "John is the tallest person who can stand up in here" may say more about the ceiling than about John, if we are talking about children in a tent.) You pays your money, you takes your choice: my construction has a good λ-calculus, whilst locales have (infinitary) algebra, which Vickers' work has exploited very fluently. Plainly we should be asking for both, and there is no fundamental reason why we shouldn't have them. The resulting category will have many more objects besides locales and colocales, and will in fact be cartesian closed. So the argument from authority (or at least from familiarity) seems to caution against a too hasty search for generality.

The applications to topology do, unfortunately, go rather quickly outside local compactness. Some account of the category of locales is therefore needed quite urgently in abstract Stone duality, and I hope that Vickers' work will help towards this. It is possible that such a construction would be the first step of an iteration that would lead to the bigger category, whilst retaining the *essence* of topology that we distill here. I believe that this lies in the preservation by $\Sigma^{\Sigma^{(-)}}$ of certain kinds of equalisers, but more liberal notions of subspace and injectivity will be needed.

Nevertheless, both domain theory and rather a lot of the corpus of pure mathematics may be seen as going on in the category of locally compact spaces, so it is well worth continuing to investigate what can be done in the abstract setting that we have so far.

2 The subspace topology

A subset $X \subset Y$ of a topological space is said to carry the ***subspace topology*** if its open subsets $U \subset X$ are exactly those of the form $U = X \cap V$ for some open subset $V \subset Y$. Since open subsets $U \subset X$ correspond bijectively to continuous maps $\phi : X \to \Sigma$, there is a more categorical way of saying this:

Remark 2.1 The Sierpiński space Σ is ***injective*** with respect to the inclusion i:

$$\begin{tikzcd} X \arrow[rr, tail, "i"] \arrow[dr, "\phi"'] & & Y \arrow[dl, dotted, "\psi"] \\ & \Sigma & \end{tikzcd}$$

for any continuous map $\phi : X \to \Sigma$, there is some continuous map $\psi : Y \to \Sigma$ such that $\phi(x) = \psi(ix)$.

In the traditional axiomatisation of general topology, the collections of open subsets of X and Y admit *arbitrary* unions, and $i^* \equiv (-) \cap X$ preserves them. This means that there is a *greatest* open $V = i_* U \subset Y$ such that $U = V \cap X$, where $i^* \dashv i_*$. In particular, if $X \subset Y$ is the closed subspace complementary to open $W \subset Y$, we have $V = U \cup W$.

As this argument manipulates open subsets and not points, it can be expressed in locale theory. We say that the locale (represented by the frame) A is a ***sublocale*** of B if there are monotone functions $i^* \dashv i_*$ such that i^* is a frame homomorphism and[2] $i_* \,; i^* = \mathsf{id}_A$. These can be recovered from the endomorphism $j = i^* \,; i_*$ on B, which satisfies $\mathsf{id}_B \leq j = j^2$ and $j(b_1 \wedge b_2) = jb_1 \wedge jb_2$. Such a j is called a ***nucleus***. In particular, the nuclei corresponding to the open subspace $W \subset Y$ and its closed complement are $W \Rightarrow (-)$ and $W \cup (-)$ respectively.

The lattice of open subsets of a space Y is the *exponential* Σ^Y in the category of locally compact spaces and continuous functions, when we equip Y with the Scott topology. In this setting, the inverse image map i^* is Σ^i. However, the *monotone* functions i_* and j need not be morphisms between the objects Σ^X and Σ^Y in this category, as, in terms of the lattices, they need not be Scott continuous.

Many of the features of general topology can be expressed in terms of the λ-calculus that arises from the exponential $\Sigma^{(-)}$, together with *finitary* lattice operations and laws [A, C]. Apart from Theorem 3.11 and some Examples, whose purpose is to motivate the ideas, the whole theory will be formulated in these abstract terms: $\mathcal{C}$ denotes a category with an internal distributive lattice Σ, of which all powers Σ^X exist in $\mathcal{C}$. In fact, even the lattice structure is not used in the main development of the present paper.

Returning to the idea of injectivity, in practice we need to state it with parameters.

Lemma 2.2 The following are equivalent for $i : X \to Y$:
(a) Σ is injective with respect to $i \times U : X \times U \to Y \times U$ for any object U;
(b) Σ^U is injective with respect to $i : X \to Y$ for any object U;
(c) there is some morphism $I : \Sigma^X \to \Sigma^Y$ such that $I \,; \Sigma^i = \mathsf{id}_{\Sigma^X}$.

$$\begin{tikzcd} X \times U \arrow[rr, tail, "i \times U"] \arrow[dr] & & Y \times U \arrow[dl, dotted] \\ & \Sigma & \end{tikzcd} \qquad \begin{tikzcd} X \arrow[rr, tail, "i"] \arrow[dr, "\phi"'] & & Y \arrow[dl, dotted, "\psi"] \\ & \Sigma^U & \end{tikzcd}$$

Proof. Parts (a) and (b) are exponential transposes of each other, and (c) is the special case of (b) with $U = \Sigma^X$ and $\phi = \eta_X$. Using (b), we obtain $I = \eta_{\Sigma^X} \,; \Sigma^\psi : \Sigma^X \to \Sigma^Y$, as

$$I \,; \Sigma^i \;=\; \eta_{\Sigma^X} \,; \Sigma^\psi \,; \Sigma^i \;=\; \eta_{\Sigma^X} \,; \Sigma^{\eta_X} \;=\; \mathsf{id}.$$

Conversely, $\psi = \eta_Y \,; \Sigma^I \,; \Sigma^{\Sigma^\phi} \,; \Sigma^{\eta_U}$ satisfies

$$i;\psi \;=\; i;\eta_Y;\Sigma^I;\Sigma^{\Sigma^\phi};\Sigma^{\eta_U} \;=\; \eta_X;\Sigma^{\Sigma^i};\Sigma^I;\Sigma^{\Sigma^\phi};\Sigma^{\eta_U} \;=\; \eta_X;\Sigma^{\Sigma^\phi};\Sigma^{\eta_U} \;=\; \phi;\eta_{\Sigma^U};\Sigma^{\eta_U} \;=\; \phi. \;\blacksquare$$

We therefore have a situation in which $i : X \rightarrowtail Y$ is a subspace (and Σ is injective with respect to it) in a very *explicit* sense, namely that the way in which open subsets $U \subset X$ are expanded to $V \subset Y$ is dictated by a morphism I.

[2]We write ";" for composition in diagrammatic order as far as Section 7, while we argue categorically. Then we switch the order and use "·" in the development of our new λ-calculus.

Definition 2.3 $i : X \to Y$ is a Σ***-split*** (mono or) ***subspace*** if there is some morphism $I : \Sigma^X \to \Sigma^Y$ such that $I\,;\Sigma^i = \mathsf{id}_{\Sigma^X}$.

$$\begin{tikzcd}
X \arrow[r, tail, "i"] & Y \\
\Sigma^X \arrow[r, tail, shift right, "I"'] & \Sigma^Y \arrow[l, two heads, shift right, "\Sigma^i"']
\end{tikzcd}$$

We shall mark Σ-split monos with a hook like this.

Remark 2.4 The idempotent $E = \Sigma^i\,; I$ on Σ^Y plays a role in our λ-calculus similar to that of the nucleus j in locale theory, except that j is uniquely determined by i, whereas E is not. (So in Section 4 we shall appropriate the name "nucleus" for E.) Beware, however, that it is not *enough* for E to be idempotent, as its epi part Σ^i must be a (frame) homomorphism. For locales, this happens iff

$$E(\phi \wedge \psi) \;=\; E(E\phi \wedge E\psi) \quad \text{and} \quad E(\phi \vee \psi) \;=\; E(E\phi \vee E\psi),$$

though we shall characterise E by a different (λ-)equation in our abstract setting.

Σ^Y is a continuous lattice, whilst E is Scott-continuous, so its splitting Σ^X is also a continuous lattice, and the space X is therefore also locally compact.

Examples 2.5 If a localic nucleus j is Scott-continuous, then it can serve as E. Otherwise, E may still exist, no longer providing the *largest* V, and the map I may extend open subsets from X to Y in very complicated ways. Some sublocales need not, however, be represented at all in our formulation.

(a) In both approaches, $I = i_*$ and $E = j = i^*\,; i_* = (- \vee W)$ encode a *closed* subspace of Y, namely that complementary to the *open* subspace W.

(b) In locale theory, $j = (W \Rightarrow -)$ encodes the open subspace W, but j is not Scott continuous unless W is also closed. Nevertheless, there is another, and rather more obvious, way of "extending" the open subset $U \subset X$ to one of Y, namely as U itself. This is of course the smallest V that does the job, so $U \mapsto V$ is the *left* adjoint $i_! \dashv i^*$. Then $I = i_!$ and $E = i^*\,; i_! = (- \wedge W)$ provide our encoding.

(c) If $X \xrightarrow{i_1} Y \xrightarrow{i_2} Z$ are successively Σ-split subspaces, with $\Sigma^X \xrightarrow{I_1} \Sigma^Y \xrightarrow{I_2} \Sigma^Z$, then the composite $X \xrightarrow{i_1;i_2} Z$ is Σ-split, with $\Sigma^X \xrightarrow{I_1;I_2} \Sigma^Z$.

(d) Hence locally closed subspaces (open subsets of closed subsets) may be also expressed in this way.

Compact and overt subspaces are considered in [E–].

Example 2.6 In set theory, subspaces and open subspaces are the same thing, but we can still distinguish two *roles* in the preceding discussion. Consider a predicate $\phi \in \Omega^X$, which classifies the "open" subset $U = \{x : X \mid \phi[x]\} \subset X$. This may be expanded from the "subspace" X to Y using either of the quantifiers:

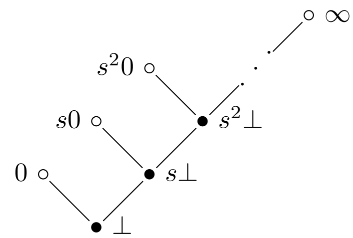
(a) $I_\exists\phi \;=\; \exists_i\phi \;=\; \lambda y{:}Y.\ \exists x{:}X.\ (y =_Y ix) \wedge \phi[x]$, or

(b) $I_\forall\phi \;=\; \forall_i\phi \;=\; \lambda y{:}Y.\ \forall x{:}X.\ (y =_Y ix) \Rightarrow \phi[x]$.

As for open and closed subspaces, $\exists_i$ and $\forall_i$ are the left and right adjoints respectively to the inverse image map i^*. This analogy between open subspaces and the existential quantifier, and between closed subspaces and the universal quantifier, is explored in [C].

The (Scott) topology on an object Y of **Dcpo** is determined by its order, and is also the exponential Σ^Y in that category. However, for a subset $X \subset Y$ that is closed under directed joins, the Scott topology on X derived from the restricted order need not be the subspace topology.

Example 2.7 (Moggi) Let Y be the domain of lazy natural numbers, and $X \subset Y$ the subset of maximal points $\circ$ in it, *i.e.* the numerals $s^n 0$ together with the "top" point ∞, omitting the points $\bullet$ of the form $s^n \bot$.



X is the equaliser of the (continuous, recursive) functions $f, g : Y \rightrightarrows Y$ for which $f(s^n 0) = g(s^n 0) = s^{n+1} 0$ but $f(s^n \bot) = s^{n+1} \bot$ and $g(s^n \bot) = s^n \bot$.

The order on X is discrete, as therefore is its *Scott* topology.

However, its *subspace* topology is that of the one-point compactification of $\mathbb{N}$: any open subset $U \subset X$ with $\infty \in U$ is of the form

$$U = \big(\{s^m 0 \mid m \geq n\} \cup \{\infty\}\big) \cup F \qquad \text{with} \quad F \subset \{s^m 0 \mid m < n\}$$

for some n. Then $V = \{y \in Y \mid y \geq s^n \bot\} \cup F$ is open in Y with $U = X \cap V$. Choosing the least such n, the assignment $I : U \mapsto V$ is monotone, so it is Scott-continuous as there are no non-trivial directed joins to be considered. Hence $X \subset Y$ is Σ-split, classically. For intuitionistic locale theory, recursion theory and our λ-calculus, the subspace is still an equaliser but the map I is not well defined. ∎

Remark 2.8 The notion of Σ-split subspace has a direct impact on computation in the λ-calculus associated to the exponential Σ^X. In any category where it is meaningful, evaluation $\mathsf{ev} : \Sigma^X \times X \to \Sigma$ is always *dinatural* in X with respect to any map $i : X \to Y$ [ML71, Section IX 4], *i.e.* the square from $\Sigma^Y \times X$ to Σ commutes:

$$\begin{tikzcd}
\Sigma^Y \times X \arrow[r, two heads, shift left, "\Sigma^i \times \mathsf{id}"] \arrow[d, "\mathsf{id} \times i"'] & \Sigma^X \times X \arrow[l, tail, shift left, "I \times \mathsf{id}"] \arrow[d, "\mathsf{ev}_X"] \\
\Sigma^Y \times Y \arrow[r, "\mathsf{ev}_Y"] & \Sigma
\end{tikzcd}$$

Combining the splitting I with this property, ev_X is equal to the composite around the other three sides, or, in symbols,

$$\phi : \Sigma^X,\ x : X \ \vdash\ \phi x \ = \ (I\phi)(ix). \tag{η}$$

This equation will later feature as an η-rule of a new λ-calculus for subspaces, based on the axiom of comprehension in set theory. The associated introduction and elimination rules are provided by Σ^i and I, whilst the β-rule expresses the composite $\Sigma^i \,; I = E$:

$$\psi : \Sigma^Y \ \vdash\ I\big(\lambda x.\, \psi(ix)\big) \ = \ E\psi. \tag{β}$$

The importance of the η-rule is that it enables us to evaluate a function ϕ on a value x in the subspace X by regarding them both as belonging to the ambient space Y instead. Therefore we may *reason mathematically* in a rich category of subspaces, but *execute the computation* using the types of the restricted λ-calculus, simply ignoring the new connectives of the comprehension calculus (Section 10).

Remark 2.9 The equation $I\,;\Sigma^i = \mathsf{id}$ says what we require of the *open subsets* of X and Y, but does not in fact even force i to be mono on *points*. We would actually like it to be *regular* mono, *i.e.* for there to be an equaliser

$$\begin{tikzcd} X \arrow[r, tail, "i"] & Y \arrow[r, shift left, "u"] \arrow[r, shift right, "v"'] & Z. \end{tikzcd}$$

The corresponding effect on the topologies is

$$\begin{tikzcd} \Sigma^X \arrow[r, shift right, tail, "I"'] & \Sigma^Y \arrow[l, shift right, two heads, "\Sigma^i"'] \arrow[r, dotted, "J"] & \Sigma^Z, \arrow[l, shift right=2, "\Sigma^u"'] \arrow[l, shift left=2, "\Sigma^v"] \end{tikzcd}$$

where the fifth map J will be explained in the next section.

Any Σ-split subspace may be expressed in this form, with

$$Z = \Sigma^{\Sigma^Y} \equiv \Sigma^2 Y \qquad u = \eta_Y \qquad v = \eta_Y\,;\Sigma^I\,;\Sigma^{\Sigma^i} \qquad J = \eta_{\Sigma^Y}.$$

Here the natural transformation η_Y is given by the λ-expression $y \mapsto \lambda\psi.\,\psi y$ and is the unit of the adjunction $\Sigma^{(-)} \dashv \Sigma^{(-)}$. Iteration of the functor $\Sigma^{(-)}$ obliges us to use the shorthand $\Sigma^2 Y$.

We have, however, been making an assumption about the category.

Definition 2.10 The maps $i = \eta_X : X \to Y = \Sigma^{\Sigma^X}$ and $I = \eta_{\Sigma^X}$ always satisfy $I\,;\Sigma^i = \mathsf{id}$ (we call the equation $\eta_{\Sigma^X}\,;\Sigma^{\eta_X} = \mathsf{id}$ the ***unit law***), so X ought to be a subspace of $\Sigma^2 X$. The diagram above becomes

$$\begin{tikzcd} X \arrow[r, tail, "\eta_X"] & \Sigma^2 X \arrow[r, shift left, "\eta_{\Sigma^2 X}"] \arrow[r, shift right, "\Sigma^2 \eta_X"'] & \Sigma^4 X \end{tikzcd}$$

with $J = \eta_{\Sigma^3 X}$. Recall from Section A 4 that an object X for which this is an equaliser is called ***sober***. Definition 2.3 and the ensuing discussion only make sense if we assume that *all* objects of $\mathcal{C}$ have this property.

A map $P : U \to \Sigma^2 X$ that has equal composites with this pair is called ***prime***, and we write $\mathsf{focus}\,P : U \to X$ for the mediator to the equaliser. Equivalently, the double exponential transpose $H : \Sigma^X \to \Sigma^U$ is an Eilenberg–Moore homomorphism, the algebra structures being Σ^{η_X} and Σ^{η_Y}. Section A 8 developed the λ-calculus with focus, and in particular $\mathsf{focus}(\lambda\phi.\,\phi a) = a$.

Notation 2.11 We shall also make use of maps $J : \Sigma^X \to \Sigma^U$ that need not be homomorphisms. We write $\widehat{J} : U \longrightarrow\!\!\!\!\times\, X$ as if this were a "second class" map between these objects, and $\mathsf{H}\mathcal{C}$ for the category with these morphisms, but the same objects as $\mathcal{C}$. For example, we may now write the equation for a Σ-split subspace as $i\,;\widehat{I} = \mathsf{id}_X$.

We define $\mathsf{force}_X \equiv \widehat{\eta_{\Sigma^X}} : \Sigma^2 X \longrightarrow\!\!\!\!\times\, X$, which is natural in $\mathsf{H}\mathcal{C}$ and (by the unit law) satisfies $\eta_X\,;\mathsf{force}_X = \mathsf{id}_X$.

If $H : \Sigma^X \to \Sigma^U$ is a homomorphism, then we write $\widehat{H} : U \to X$, with an ordinary arrowhead instead of $\longrightarrow\!\!\!\!\times$, and $\mathsf{S}\mathcal{C}$ for the category with these morphisms. When all objects are sober, $\mathsf{S}\mathcal{C} \cong \mathcal{C}$, *i.e.* any such $\widehat{H} : U \to X$ is of the form $H = \Sigma^f$ for some unique ordinary morphism $f : U \to X$ in $\mathcal{C}$, and we write $f : U \to X$ instead of $\widehat{\Sigma^f}$.

In fact, $\widehat{H} = \mathsf{focus}\,P = P\,;\mathsf{force}_X$, where the prime P is the double exponential transpose of H (Lemma A 7.5). The reason for the distinction between focus and force is that the former is part of a *denotational* calculus, whilst the latter introduces "computational effects".

Example 2.12 When $\mathcal{C} = \mathbf{LKLoc}$ is the category of locally compact locales, $\mathsf{S}\mathcal{C}$ is the opposite of the category of (continuous) frames and frame homomorphisms, so $\mathsf{S}\mathcal{C} \cong \mathcal{C}$. $H : U \longrightarrow\!\!\!\!\times\, X$ denotes a Scott-continuous map from the topology on X to that on U. ∎

3 Beck's theorem

Now we relate subspaces as we have just described them to Stone duality as presented in Section 1. In the abstract setting, the diagrams shown there are as follows, where **Alg** is the category of Eilenberg–Moore algebras for the monad induced by the adjunction $\Sigma^{(-)} \dashv \Sigma^{(-)}$.

$$\begin{tikzcd}
\mathcal{C}^{\mathsf{op}} \arrow[rr, shift left, "X \mapsto (\Sigma^X,\, \Sigma^{\eta_X})"] \arrow[rr, phantom, "\simeq"] \arrow[d, shift left, "\Sigma^{(-)}"] \arrow[d, phantom, "\dashv"] & & \mathbf{Alg} \arrow[ll, shift left, "\mathsf{pts}"] \arrow[d, shift left, "(A,\alpha) \mapsto A"] \arrow[d, phantom, "\dashv"] \\
\mathcal{C} \arrow[u, shift left, "\Sigma^{(-)}"] \arrow[rr, equal] & & \mathcal{C} \arrow[u, shift left, "X \mapsto (\Sigma^{\Sigma^X},\ \mu_X)"]
\end{tikzcd}$$

We shall show that subspaces (*i.e.* certain equalisers) exist in $\mathcal{C}$ and carry the subspace topology iff (idempotents split and) the functor $\mathcal{C}^{\mathsf{op}} \to \mathcal{A}$ is an equivalence of categories. This functor is full and faithful iff all objects are sober (Theorem A 4.10), so our task in this paper is to say when it is essentially surjective. In fact we shall construct the pseudo-inverse functor. The argument works though the proof Beck's theorem in our special case.

Definition 3.1 An algebra (A, α) is ***spatial*** if there is some algebra isomorphism

$$\begin{tikzcd}
\Sigma^2 A \arrow[r, "\Sigma^2 H"] \arrow[d, "\alpha"'] & \Sigma^3 X \arrow[d, "\Sigma^{\eta_X}"] \\
A \arrow[r, "H"] & \Sigma^X
\end{tikzcd}$$

Recall from Lemma A 4.3 that a map $H : A \to \Sigma^U$ (not necessarily an isomorphism) is a homomorphism iff its double exponential transpose P has equal composites

$$\begin{tikzcd}
U \arrow[r, "P"] & \Sigma^A \arrow[r, shift left, "\Sigma^\alpha"] \arrow[r, shift right, "\eta_{\Sigma A}"'] & \Sigma^3 A,
\end{tikzcd}$$

so our first attempt at spatiality of (A, α) is to *form* this equaliser [Fak70]. Afterwards we have to find out when H is an isomorphism.

Proposition 3.2 The contravariant functor $X \mapsto (\Sigma^X, \Sigma\eta_X) : \mathcal{C}^{\mathsf{op}} \to \mathcal{A}$ has a symmetric adjoint on the right, called pts, so

$$\frac{X \longrightarrow \mathsf{pts}(A,\alpha) \text{ in } \mathcal{C}}{A \longrightarrow \Sigma^X \text{ in } \mathbf{Alg}}$$

iff for each algebra (A, α) the equaliser

$$\begin{tikzcd}
\mathsf{pts}(A,\alpha) \arrow[r, "i"] & \Sigma^A \arrow[r, shift left, "\Sigma^\alpha"] \arrow[r, shift right, "\eta_{\Sigma A}"'] & \Sigma^3 A
\end{tikzcd}$$

exists in $\mathcal{C}$.

Proof. By the foregoing argument, there must also be a correspondence with maps

$$\begin{tikzcd}
X \arrow[r] & \Sigma^A \arrow[r, shift left] \arrow[r, shift right] & \Sigma^3 A,
\end{tikzcd}$$

but this is just the universal property of the equaliser. ∎

The inverse functor $\mathsf{pts} : \mathbf{Alg} \to \mathcal{C}^{\mathsf{op}}$ takes the algebra A to the "set" $\mathbf{Alg}(A, \Sigma)$ of homomorphisms, just as the forward one took the object X to the function-space $\mathcal{C}(X, \Sigma)$. (Remark 7.4 explains how $\mathbf{Alg}(A, B)$ can sometimes be defined as an object of $\mathcal{C}$, rather than as an external hom-set.)

Theorem 3.3 The contravariant adjunction of $\Sigma^{(-)}$ and pts is an equivalence between $\mathcal{C}^{\mathsf{op}}$ and $\mathbf{Alg}$ iff all objects of $\mathcal{C}$ are sober and all algebras in $\mathbf{Alg}$ are spatial.

Proof. Sobriety and spatiality merely say that the units of the symmetric adjunction,

$$X \to \mathsf{pts}(\Sigma^X, \Sigma^{\eta_X}) \qquad \text{and} \qquad H : A \to \Sigma^{\mathsf{pts} A},$$

are isomorphisms. ∎

More particularly, without assuming that all objects are sober, the functor $\mathsf{pts} : \mathbf{Alg}^{\mathsf{op}} \to \mathcal{C}$ is full and faithful (making $\mathbf{Alg}^{\mathsf{op}}$ reflective in $\mathcal{C}$) iff all algebras are spatial. Joachim Lambek and Basil Rattray considered this situation abstractly (writing QA for our $\Sigma^{\mathsf{pts} A}$), together with applications to Abelian categories and elementary toposes [LR75]. More recently, Giuseppe Rosolini connected their results with topos models of synthetic domain theory [BR98].

We can characterise spatiality in terms more like those of the previous section.

Proposition 3.4 $(A, \alpha) \cong \Sigma^X$ iff there are $i : X \to \Sigma^A$ and $I : \Sigma^X \to \Sigma^2 A$ such that

$$i \text{ is prime}, \qquad I\,;\Sigma^i = \mathsf{id}_{\Sigma^X} \quad \text{and} \quad \Sigma^i\,;I = \alpha\,;\eta_A \equiv \eta_{\Sigma^2 A}\,;\Sigma^2\alpha.$$

In this case, $X = \mathsf{pts}(A, \alpha) \xrightarrow{i} \Sigma^A$ is the equaliser in Proposition 3.2, and $\Sigma^{(-)}$ takes it to a split coequaliser.

$$\begin{tikzcd}
A \arrow[r, shift left, "\eta_A"] \arrow[d, dotted, "H"'] & \Sigma^2 A \arrow[l, shift left, "\alpha"] \arrow[dl, "\Sigma^i"] \arrow[r, shift right=2, "\eta_{\Sigma^2 A}"'] & \Sigma^4 A \arrow[l, shift right=2, "\Sigma^2\alpha"'] \arrow[l, "\Sigma\eta_{\Sigma A}" description] \\
\Sigma^X & &
\end{tikzcd}$$

Proof. If $H : A \cong \Sigma^X$ is an isomorphism of algebras then its double exponential transpose $i = \eta_X\,;\Sigma^H$ is prime, and we may put $I = H^{-1}\,;\eta_A = \eta_{\Sigma^X}\,;\Sigma^2 H^{-1}$. Then

$$I\,;\Sigma^i \;=\; \eta_{\Sigma^X}\,;\Sigma^2 H^{-1}\,;\Sigma^2 H\,;\Sigma^{\eta_X} \;=\; \eta_{\Sigma^X}\,;\Sigma^{\eta_X} \;=\; \mathsf{id}$$

$$\Sigma^i\,;I \;=\; \Sigma^2 H\,;\Sigma^{\eta_X}\,;H^{-1}\,;\eta_A \;=\; \alpha\,;H\,;H^{-1}\,;\eta_A \;=\; \alpha\,;\eta_A.$$

Conversely, if i is prime then its double transpose $H = \eta_A\,;\Sigma^i$ is a homomorphism, and $H^{-1} = I\,;\alpha$ because

$$\eta_A\,;\Sigma^i\,;I\,;\alpha \;=\; \eta_A\,;\alpha\,;\eta_A\,;\alpha \;=\; \mathsf{id}$$

$$I\,;\alpha\,;\eta_A\,;\Sigma^i \;=\; I\,;\eta_{\Sigma^2 A}\,;\Sigma^2\alpha\,;\Sigma^i \;=\; I\,;\eta_{\Sigma^2 A}\,;\Sigma^{\eta_{\Sigma^A}}\,;\Sigma^i \;=\; I\,;\Sigma^i \;=\; \mathsf{id}$$

since i, being prime, has equal composes with Σ^α and η_{Σ^A}.

To show that X is the equaliser, consider $\Gamma \xrightarrow{P} \Sigma^A \rightrightarrows \Sigma^3 A$. The double transpose $K : A = \Sigma^X \to \Sigma^\Gamma$ of P is a homomorphism, so, since X is sober, $K = \Sigma^k$, where $k : \Gamma \to X$ mediates to the equaliser. The image of the equaliser diagram under $\Sigma^{(-)}$ is the split coequaliser shown above. ∎

Remark 3.5 In the λ-calculus, these equations are

$$x : X,\ \phi : \Sigma^X \ \vdash\ (I\phi)(ix) \ =\ \phi(x) \tag{η}$$
$$F : \Sigma^A,\ \mathcal{F} : \Sigma^2 A \ \vdash\ I\big(\lambda x.\, \mathcal{F}(ix)\big)(F) \ =\ F(\alpha\mathcal{F}). \tag{β}$$

The η-equation, which says that Σ^X is a retract of A, was discussed in the previous section: it makes X a Σ-split subspace of Σ^A.

The β-equation makes $H : A \rightarrowtail \Sigma^X$ a subalgebra that is "U-split" in Beck's language, *i.e.* split by a function that need not be a homomorphism. This property says that there are ***enough points*** to distinguish the elements of A, considered as "open subsets" of X.

Remark 3.6 The same caveat applies to our definition of "spatial" algebras as to that of "sober" spaces in Remark A 5.10: the correspondence with the lattice-theoretic notion in [Joh82, Section II 1.5] is a conceptual one rather than a theorem. In particular, recall from [Joh82, Theorem VII 4.3] that (using the axiom of choice) *any* distributive continuous lattice A has enough points (completely coprime filters) in the classical sense. All algebras over **LKLoc** are spatial in the constructive sense that there are enough *generalised* "points" $\Gamma \to \mathsf{pts}(A, \alpha)$, *cf.* [Vic95].

Beck's theorem and our treatment of subspaces consider the *maps and equations* in the main equaliser diagram, without requiring the *objects* to be $\Sigma^2 A$, $\Sigma^3 X$, *etc.*

Definition 3.7 A parallel pair $u, v : Y \rightrightarrows Z$ in $\mathcal{C}$ is called a Σ-***split pair*** if there is some map (not necessarily a homomorphism) $J : \Sigma^Y \to \Sigma^Z$ such that

$$J \,;\, \Sigma^u \ =\ \mathsf{id}_{\Sigma^Y} \qquad \text{and} \qquad \Sigma^u \,;\, J \,;\, \Sigma^v \ =\ \Sigma^v \,;\, J \,;\, \Sigma^v.$$

Notice that we just mark one of the maps with a hook, to emphasise the fact that the conditions are not symmetrical in u and v.

$$\begin{tikzcd} \Sigma^Y \arrow[r, tail, "J" description] & \Sigma^Z \arrow[l, two heads, bend right, "\Sigma^u"'] \arrow[l, bend left, "\Sigma^v"] \end{tikzcd} \qquad \begin{tikzcd} Y \arrow[rr, shift left, "u"] \arrow[rr, shift right, "v"'] \arrow[dr, "\psi"'] & & Z \arrow[dl, "\theta"] \\ & \Sigma & \end{tikzcd}$$

$$\begin{tikzcd} Y \arrow[r, bend left, "u"] \arrow[r, bend right, "v"'] & Z \arrow[l, "\widehat{J}" description] \end{tikzcd} \qquad \psi \overset{J}{\longmapsto} \theta$$

It is helpful to see these equations expressed in other ways. Using Notation 2.11, they are

$$u \,;\, \widehat{J} \ =\ \mathsf{id} \quad \text{and} \quad v \,;\, \widehat{J} \,;\, u \ =\ v \,;\, \widehat{J} \,;\, v.$$

Mixing application with categorical composition, we have

$$u \,;\, J(\psi) \ =\ \psi \qquad \text{and} \qquad v \,;\, J(u \,;\, \theta) \ =\ v \,;\, J(v \,;\, \theta).$$

Using the λ-calculus, the equations are

$$\begin{aligned} y : Y,\ \psi : \Sigma^Y \ &\vdash\ (J\psi)(uy) \ =\ \psi(y) \\ y : Y,\ \theta : \Sigma^Z \ &\vdash\ J\big(\lambda y.\, \theta(uy)\big)(vy) \ =\ J\big(\lambda y.\, \theta(vy)\big)(vy). \end{aligned}$$

We shall need the ***equaliser*** $i : X \to Y$ of u and v, for which we would like $\Sigma^i : \Sigma^Y \to \Sigma^X$ also to be the *co*equaliser of Σ^u and Σ^v. In this case there is a (unique) map $I : \Sigma^X \to \Sigma^Y$ with $I \,;\, \Sigma^i = \mathsf{id}_{\Sigma^X}$ and $\Sigma^i \,;\, I = J \,;\, \Sigma^v$.

Reversing the arrows, there is a similar notion of Σ-split coequaliser (Section 11), whilst (unqualified) split equalisers and coequalisers also arise, in which the equations already hold at the base level, without applying the functor $\Sigma^{(-)}$. Split (co)equalisers are ***absolute*** in the sense that, being equationally defined, they are preserved by *all* functors. Absolute coequalisers were first studied by Robert Paré [Par71]. Examples of absolute *pushouts* may be found in Section 7, [Tay99, Exercise 5.3] and [D].

We *construct* the equaliser of a split pair by splitting an idempotent (Remark 4.1). In particular, Σ^X splits $E = \Sigma^v \,;\, J$. Conversely, any idempotent $e : Y \to Y$ gives rise to a split pair (e, id), with $J = E = \Sigma^e$, so such splittings are necessary.

There is no greater generality in supposing that i is split only after further iteration of the functor $\Sigma^{(-)}$, since if $\mathcal{I} \,;\, \Sigma^3 i = \mathsf{id}$ then $I \,;\, \Sigma^i = \mathsf{id}$ by the unit law, where $I = \eta_{\Sigma^X} \,;\, \mathcal{I} \,;\, \Sigma^{\eta_Y}$.

Proposition 3.8 If the functor pts exists, all objects are sober and idempotents split, then $\mathcal{C}$ has equalisers of *all* Σ-split pairs. If, additionally, all algebras are spatial, $\Sigma^{(-)}$ takes these equalisers to (split) coequalisers.

Conversely, if $\mathcal{C}$ has equalisers of all Σ-split pairs, and $\Sigma^{(-)}$ takes them to coequalisers, then all algebras are spatial.

Proof. Let $u, v : Y \rightrightarrows Z$ have splitting $J : \Sigma^Z \to \Sigma^Y$, so that $E = \Sigma^v \,;\, J$ is an idempotent on Σ^Y, and let A be its splitting, as in the top rows of the diagrams. Since this coequaliser is absolute, $\Sigma^2 A$ is also a coequaliser, so the structure α may be obtained as the mediator; similar arguments involving $\Sigma^4 A$ show that it satisfies the equations for an Eilenberg–Moore algebra.

$$
\begin{tikzcd}
\mathsf{pts}(A,\alpha) \arrow[r, dotted] \arrow[d, "i"'] & Y \arrow[r, shift left, "u"] \arrow[r, shift right, "v"'] \arrow[d, "\eta_Y"] & Z \arrow[d, "\eta_Z"] \\
\Sigma^A \arrow[r, "\Sigma^q"] \arrow[d, shift right, "\Sigma^\alpha"'] \arrow[d, shift left, "\eta_{\Sigma^A}"] & \Sigma^2 Y \arrow[r, shift left, "\Sigma^2 u"] \arrow[r, shift right, "\Sigma^2 v"'] \arrow[d, shift right] \arrow[d, shift left] & \Sigma^2 Z \arrow[d, shift right] \arrow[d, shift left] \\
\Sigma^3 A \arrow[r] & \Sigma^4 Y \arrow[r, shift left] \arrow[r, shift right] & \Sigma^4 Z
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[r, shift right, "I"'] & \Sigma^Y \arrow[l, shift right, "q"'] \arrow[r, "J"] & \Sigma^Z \arrow[l, shift right=2, "\Sigma^u"'] \arrow[l, shift left=2, "\Sigma^v"] \\
\Sigma^2 A \arrow[u, "\alpha"] & \Sigma^3 Y \arrow[l] \arrow[u, "\Sigma^{\eta_Y}"] & \Sigma^3 Z \arrow[l, shift left] \arrow[l, shift right] \arrow[u, "\Sigma^{\eta_Z}"']
\end{tikzcd}
$$

I claim that $\mathsf{pts}(A, \alpha)$ is the equaliser of u and v.

All composites $\mathsf{pts}(A, \alpha) \to \Sigma^4 Y$ are equal, so they factor through η_Y since Y is sober. This defines the dotted map. It has equal composites with u and v because all composites $\mathsf{pts}(A, \alpha) \to \Sigma^2 Z$ are equal and η_Z is mono.

Any test map $\Gamma \to Y$ having equal composites with u and v also has equal composites with $\Sigma^2 u$ and $\Sigma^2 v$, since η is natural, so it factors through their equaliser, Σ^A. For the same reasons it has equal composites with $\Sigma^A \rightrightarrows \Sigma^3 A$, and so factors through their equaliser $\mathsf{pts}(A, \alpha)$. The mediator is unique because i and Σ^q are mono.

Now apply $\Sigma^{(-)}$ to the left-hand diagram, so $\Sigma^{\mathsf{pts}(A,\alpha)} \cong A$ and $\Sigma^i \cong \alpha$ by Theorem 3.3 since A is spatial. Then $\Sigma^{(-)}$ takes the top two rows of the diagram on the left to the one on the right.

Since pts is itself a Σ-split equaliser, the converse follows from Proposition 3.4. ∎

Remark 3.9 Because of the fact that $\Sigma^{(-)}$ takes it to the coequaliser of $\Sigma^Y \leftleftarrows \Sigma^Z$, the diagram $X \to Y \rightrightarrows Z$ is also an equaliser in $\mathsf{H}\mathcal{C}$, *i.e.* with respect to "second class" test maps $\widehat{F} : \Gamma \longrightarrow\!\!\!\!\!\times\; Y \rightrightarrows Z$ (Notation 2.11).

This is in contrast to the situation for products, where the failure of the extension of the universal property from $\mathcal{C}$ to $\mathsf{H}\mathcal{C}$ is essentially related to the interpretation of computational effects [Thi97]. The fact that equalisers are valid even in the presence of such effects may be regarded as the categorical justification of the use of Floyd–Hoare logic for imperative as well as functional programs.

It is still not clear precisely what the exactness properties of $\Sigma^{(-)}$ should be, as not *all* equalisers are taken to coequalisers:

Example 3.10 Suppose that $\mathcal{C}$ has disjoint coproducts, as is the case for **Set** and **LKSp**, and also in the abstract situation by Theorem 11.8. Then

$$\begin{tikzcd} \mathbf{0} \arrow[r, tail] & \mathbf{2} \arrow[r, shift left, "\mathsf{id}"] \arrow[r, shift right, "\mathsf{swap}"'] & \mathbf{2}, \end{tikzcd}$$

is an equaliser (where swap interchanges the *elements* of $\mathbf{2}$), but $\Sigma^{(-)}$ takes it to

$$\begin{tikzcd} \mathbf{1} & \Sigma \arrow[l] & \Sigma\times\Sigma \arrow[l, two heads] & \Sigma\times\Sigma, \arrow[l, shift right, "\mathsf{id}"'] \arrow[l, shift left, "\mathsf{swap}"] \end{tikzcd}$$

where swap interchanges the *components* of the product. ■

We can now prove the intuitionistic form of Stone duality for locally compact locales.

Theorem 3.11 **LKLoc** is monadic.

Proof. We must show that every object is sober, that equalisers of Σ-split pairs exist, and that $\Sigma^{(-)}$ takes them to coequalisers. Recall from the theory of locales and continuous lattices [GHK+80, Joh82] that the topology on a locally compact locale is a distributive continuous lattice, whilst the category **Cont** of continuous lattices and Scott continuous functions is fully embedded (*via* the Scott topology) as a subcategory of **LKLoc** that is closed under retracts.

$$\begin{tikzcd} \Sigma^X \arrow[r, shift right] & \Sigma^3 X \arrow[l, two heads, shift right] \arrow[r, shift right=2] & \Sigma^5 X \arrow[l, two heads, shift right=2] \arrow[l] \end{tikzcd}$$

The equaliser diagram of locales involved in the definition of sobriety gives rise to diagram of frames shown above, where $\twoheadleftarrow$ denotes a frame homomorphism and $\rightarrow$ a Scott continuous function. This is a split coequaliser in **Cont** and so *ā fortiori* a coequaliser in **LKFrm**, *i.e.* an equaliser in **LKLoc**.

$$\begin{tikzcd} \mathbf{LKLoc}^{\mathsf{op}} \arrow[r, equal] \arrow[d, shift left, "\Sigma^{(-)}"] & \mathbf{LKFrm} \arrow[r, tail] \arrow[d, "U'"] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbf{Frm} \arrow[d, shift left, "U"] \\ \mathbf{LKLoc} \arrow[u, shift left, "\Sigma^{(-)}"] \arrow[u, phantom, "\dashv"] & \mathbf{Cont} \arrow[l, "\Sigma(-)"'] \arrow[r] & \mathbf{Set} \arrow[u, shift left, "F"] \arrow[u, phantom, "\dashv"] \end{tikzcd}$$

Now let $u, v : Y \rightrightarrows Z$ be a Σ-split pair in **LKLoc**. We follow it around the diagram of categories and functors above.

$E = J\,;\Sigma^v$ is a Scott continuous idempotent on the continuous frame Σ^Y, so it is split by some continuous lattice Q. Hence

$$\begin{tikzcd} \Sigma^Z \arrow[r, two heads, shift left=2, "\Sigma^u"] \arrow[r, "\Sigma^v"description] & \Sigma^Y \arrow[l, shift left=2, "J"] \arrow[r, dotted, two heads, shift left, "q"] & Q \arrow[l, dotted, shift left, "I"] \end{tikzcd}$$

is a split coequaliser diagram in **LKLoc**, although we may just as well see it as being in **Cont** or **Set**.

The forgetful functor $U : \mathbf{Frm} \to \mathbf{Set}$, being monadic [Joh82, Theorem II 1.2], creates this coequaliser in **Frm**, so the lattice Q is actually a frame, and the map $\Sigma^Y \twoheadrightarrow Q$ is a frame homomorphism. This means that $Q \cong \Sigma^X$ for some locale X, which is locally compact since Q is continuous, and $q = \Sigma^i$. Hence the diagram is a coequaliser in **LKFrm**, so X is the equaliser in **LKLoc**, which $\Sigma^{(-)}$ takes to the original split coequaliser. ∎

We conclude our introductory discussion with an example of a non-spatial algebra.

Classically, the two-point set classifies subsets in **Set**, open subsets in **LKSp** (when it is given the Sierpiński topology), and upper subsets in **Pos** (when it is given the order $\bot < \top$). The monad may therefore be defined on **Pos** using this object, which we call Υ.

Example 3.12 In (**Pos**, Υ), all objects are sober classically but not intuitionistically (Remark C 5.10(c)). **Pos** *does not* have the monadic property, because the unit interval, $A = [0, 1]$ with the usual arithmetical order, is an Eilenberg–Moore algebra (in a unique way) that does not have enough points.

This discussion of $[0, 1]$ extends that in [FW90, Example 9]. In fact, the algebras for this version of the monad are constructively completely distributive lattices [MRW01].

Proof. Classically, the upper sets of A are of the form $(a, 1]$ or $[a, 1]$. Indeed $\Sigma^{A\cdot n} = A^{\mathsf{op}}\cdot(n+1)$, where $A \cdot n$ is $A \times \mathbf{n}$ with the lexicographic order: $\langle a, i\rangle \le \langle b, j\rangle$ if $a < b$, or $a = b$ and $i \le j$.

$$\begin{tikzcd}[column sep=large]
\emptyset \arrow[r] & \Sigma^A \equiv A\cdot 2 \arrow[rr, shift left, "\eta_{A\cdot 2} : 0,1 \mapsto 1,2"] \arrow[rr, shift right, "\Sigma\alpha : 0,1 \mapsto 0,3"'] & & A\cdot 4 \\
A & \Sigma^{\Sigma^A} \equiv A\cdot 3 \arrow[l, two heads, "\alpha"'] & & A\cdot 5 \arrow[ll, shift right, "{2,2,1,0,0 \mapsfrom 0,1,2,3,4 : \Sigma\eta_{A\cdot 2}}"'] \arrow[ll, shift left, "{2,0,0,0,0 \mapsfrom 0,1,2,3,4 : \Sigma^2\alpha}"]
\end{tikzcd}$$

The maps we're interested in act as the identity on the real part, and as shown above on the numerical part. ∎

4 Adding Sigma-split subspaces

The rest of the paper shows (in three different ways) how to *construct* a category $\overline{\mathcal{C}}$ in which we may form subspaces as just described, *i.e.* equalisers of Σ-split pairs that $\Sigma^{(-)}$ takes to coequalisers. As raw material, we simply require a category $\mathcal{C}$ with finite products and an exponentiating object Σ, such that all objects of $\mathcal{C}$ are sober. A category $\mathcal{C}$ of this kind was defined using a λ-calculus in Section A 8. In Section 7 we shall see that $\mathbf{Alg}^{\mathsf{op}}$ does the job, but first we do it by adding formal equalisers of Σ-split pairs.

This construction is similar to that of the so-called ***Karoubi completion*** of a category, which forces it to have splittings of idempotents. However, as the new construction is rather more complicated, you would be well advised to (re)familiarise yourself with the simpler version by proving

Remark 4.1 The Karoubi completion, $\mathsf{K}\mathcal{C}$, of any category $\mathcal{C}$ has

- as *objects*, (X, e), where $e : X \to X$ is an ***idempotent***, that is, $e\,;e = e$, and
- as *morphisms* $f : (X, e_1) \to (Y, e_2)$, the $\mathcal{C}$-maps $f : X \to Y$ with $e_1\,; f = f = f\,; e_2$.

Then

(a) composition is inherited from $\mathcal{C}$, but the identity on (X, e) is e, and

(b) the idempotents e' of $\mathsf{K}\mathcal{C}$ are those of $\mathcal{C}$, but they ***split*** in $\mathsf{K}\mathcal{C}$, *i.e.* there are maps p and i in $\mathsf{K}\mathcal{C}$ such that $i\,; p = \mathsf{id}$ and $p\,; i = e'$.

(c) $\mathsf{K}\mathcal{C}$ is the universal way (up to equivalence) of splitting idempotents in $\mathcal{C}$. ∎

From a categorical point of view, it would be easier to assume that idempotents split in $\mathcal{C}$. This would "modularise" the construction of $\overline{\mathcal{C}}$ as $\mathcal{C} \mapsto \mathsf{K}\mathcal{C} \mapsto \mathbf{Alg}_{\mathsf{K}\mathcal{C}} \mapsto \overline{\mathcal{C}}$. However, our intention (in Section 10) is to reduce computation in $\overline{\mathcal{C}}$ back to the restricted λ-calculus, *i.e.* $\mathcal{C}$. $\mathsf{K}\mathcal{C}$ already has some subspaces for the convenience of mathematicians, whereas the idempotents infest computations in it, so it has already crossed the line of demarcation between these subjects. For this reason, we develop a representation of $\overline{\mathcal{C}}$-objects directly in terms of $\mathcal{C}$ that splits idempotents as well as equalisers.

Definition 4.2 A typical *object* X of our category $\overline{\mathcal{C}}$ is a Σ-split pair (Definition 3.7), which we may write using Notation 2.11 simply as

$$\begin{tikzcd} Y \arrow[r, shift left=2, "u"] \arrow[r, "v" description] & Z, \arrow[l, shift left=2, "\widehat{J}", tail] \end{tikzcd}$$

such that $u\,;\widehat{J} = \mathsf{id}_Y$ and $v\,;\widehat{J}\,;u = v\,;\widehat{J}\,;v : Y \longrightarrow\!\!\!\!\times\; Z$. This object will temporarily be called $(Y \rightrightarrows Z)$, or just X, in parts of the argument up to Section 6.

The *identity morphism* on this object will be $\widehat{E} = v\,;\widehat{J}$. As in the Karoubi construction, this is an idempotent endomorphism of Y, but in the auxiliary category $\mathsf{H}\mathcal{C}$. In fact, its defining equation is stronger than idempotence.

Definition 4.3 $E : \Sigma^Y \to \Sigma^Y$ is called a ***nucleus*** on Y if $\widehat{E}\,;\eta_Y\,;\Sigma^E = \widehat{E}\,;\eta_Y$, or

$$\mathcal{F} : \Sigma^3 Y \;\vdash\; E\big(\lambda y.\, \mathcal{F}(\lambda\phi.\, E\phi y)\big) \;=\; E\big(\lambda y.\, \mathcal{F}(\lambda\phi.\, \phi y)\big)$$

using the λ-calculus. Observe carefully that the left hand side has an extra E.

We shall write $\{Y \mid E\}$ instead of $(Y \rightrightarrows Z)$ for an object of $\overline{\mathcal{C}}$ defined in terms of $\widehat{E}$ like this.

When $\widehat{E}$ is a first class map ($E = \Sigma^e$ and $e = \widehat{E}$), the equation is just idempotence ($e = e\,;e$). In general, however, it says more than $\widehat{E} = \widehat{E}\,;\widehat{E}$, for the same reason that we remarked in Definition 2.3, namely that, in the splitting of this second class idempotent as $\widehat{I}\,;i$, the inclusion part i must actually be first class.

Notation 4.4 Given two nuclei, E_1 and E_2, we write $\widehat{E_2} \subset_Y \widehat{E_1}$ if

$$\widehat{E_1}\,;\widehat{E_2} \;=\; \widehat{E_2} \;=\; \widehat{E_2}\,;\widehat{E_1}.$$

Definition 4.5 A typical *morphism* from $\{Y_1 \mid E_1\}$ to $\{Y_2 \mid E_2\}$ is an $\mathsf{H}\mathcal{C}$-map $\widehat{H} : Y_1 \longrightarrow\!\!\!\!\times\; Y_2$ such that

$$\widehat{H} \;=\; \widehat{H}\,;\widehat{E_2} \qquad \text{and} \qquad \widehat{E_1}\,;\eta_{Y_1}\,;\Sigma^H \;=\; \widehat{H}\,;\eta_{Y_2},$$

the first equation being equivalent to $\widehat{H}\,;u_2 = \widehat{H}\,;v_2$ when $\{Y_2 \mid E_2\}$ is defined by the pair $u_2, v_2 : Y_2 \rightrightarrows Z_2$. Using λ-calculus, the second equation is

$$\mathcal{G} : \Sigma^3 Y_2 \;\vdash\; E_1\big(\lambda x{:}Y_1.\, \mathcal{G}(\lambda\psi{:}\Sigma^{Y_2}.\, H\psi x)\big) \;=\; H\big(\lambda y{:}Y_2.\, \mathcal{G}(\lambda\psi.\, \psi y)\big),$$

which adds E_1 to the λ-equation for a homomorphism (Remark A 4.11).

Lemma 4.6 Any morphism $\widehat{H} : \{Y_1 \mid E_1\} \to \{Y_2 \mid E_2\}$ also satisfies $\widehat{H} = \widehat{E_1}\,;\widehat{H}$.

Proof.

$$\begin{array}{rcll} \widehat{H} & = & \widehat{H}\,;\eta_{Y_2}\,;\widehat{\eta_{\Sigma^{Y_2}}} & \text{unit law} \\ & = & \widehat{E_1}\,;\eta_{Y_1}\,;\Sigma^H\,;\widehat{\eta_{\Sigma^{Y_2}}} & \text{second } H \text{ equation} \\ & = & \widehat{E_1}\,;\eta_{Y_1}\,;\widehat{\eta_{\Sigma^{Y_1}}}\,;\widehat{H} & \text{naturality} \\ & = & \widehat{E_1}\,;\widehat{H} & \text{unit law. } \blacksquare \end{array}$$

Lemma 4.7 For any split pair, $E = J\,;\Sigma^v$ is a nucleus (Definition 4.3); in particular, $\widehat{E}$ is idempotent in $\mathsf{H}\mathcal{C}$.

PROOF. The equation for $\widehat{E}$ is the same as the second one for a morphism $\widehat{H} = \widehat{E}$ in Remark 4.5. It follows from the defining equations for u, v and J:

$$\begin{array}{rcll}
\widehat{E}\,;\eta_Y\,;\Sigma^E & = & v\,;\widehat{J}\,;\eta_Y\,;\Sigma^2 v\,;\Sigma^J & \\
& = & v\,;\widehat{J}\,;v\,;\eta_Z\,;\Sigma^J & \text{naturality of } \eta \\
& = & v\,;\widehat{J}\,;u\,;\eta_Z\,;\Sigma^J & \text{second } uvJ \text{ equation} \\
& = & v\,;\widehat{J}\,;\eta_Y\,;\Sigma^2 u\,;\Sigma^J & \text{naturality of } \eta \\
& = & v\,;\widehat{J}\,;\eta_Y \;=\; \widehat{E}\,;\eta_Y & \text{first } uvJ \text{ equation.}
\end{array}$$

The first equation for a morphism is idempotence of $\widehat{H} = \widehat{E}$, which follows from the previous result, but explicitly from the uvJ equations,

$$\widehat{E}\,;\widehat{E} \;=\; v\,;\widehat{J}\,;v\,;\widehat{J} \;=\; v\,;\widehat{J}\,;u\,;\widehat{J} \;=\; v\,;\widehat{J} \;=\; \widehat{E}. \qquad \blacksquare$$

Proposition 4.8 $\overline{\mathcal{C}}$ is a category.

PROOF. We have to show that composition is well defined. Let $\widehat{H} : Y_1 \to Y_2$ and $\widehat{K} : Y_2 \to Y_3$. Then $\widehat{H}\,;\widehat{K}\,;\widehat{E_3} = \widehat{H}\,;\widehat{K} = \widehat{K;H}$ and

$$\begin{array}{rcll}
\widehat{E_1}\,;\eta_{Y_1}\,;\Sigma^{K;H} & = & \widehat{E_1}\,;\eta_{Y_1}\,;\Sigma^H\,;\Sigma^K & \\
& = & \widehat{H}\,;\eta_{Y_2}\,;\Sigma^K & \text{second } H \text{ equation} \\
& = & \widehat{H}\,;\widehat{E_2}\,;\eta_{Y_2}\,;\Sigma^K & \text{first } H \text{ equation} \\
& = & \widehat{H}\,;\widehat{K}\,;\eta_{Y_3} & \text{second } K \text{ equation,}
\end{array}$$

so $\widehat{H}\,;\widehat{K}$ satisfies the definition of a morphism. Associativity is inherited from $\mathcal{C}$, whilst the equations that say that $\widehat{E}$ is the identity on $\{Y \mid E\}$ are those in Definition 4.5 and Lemma 4.6. $\blacksquare$

Remark 4.9 Although there is more to the data for an object of $\overline{\mathcal{C}}$ than the maps u and v, the justification of calling such an object a "pair" is that, if we have two splittings J_1 and J_2, then $\widehat{E_1} = v\,;\widehat{J_1}$ and $\widehat{E_2} = v\,;\widehat{J_2}$ define an isomorphism between these objects of $\overline{\mathcal{C}}$. For example, we shall see in Proposition 5.12 that the binary product of any two non-trivial objects has two different splittings.

Lemma 4.10 $\mathcal{C}$ is embedded as a full subcategory of $\overline{\mathcal{C}}$ by

$$X \quad\mapsto\quad \{X \mid \mathsf{id}\} \quad\equiv\quad \begin{tikzcd} X \arrow[r, shift left=2, "\mathsf{id}"] \arrow[r, "\mathsf{id}" description] & X \arrow[l, shift left=2, "\mathsf{id}"] \end{tikzcd}$$

$$f : X \to Y \quad\mapsto\quad H = \Sigma^f : \Sigma^Y \to \Sigma^X.$$

(Recall that we assume that all objects of $\mathcal{C}$ are sober.) $\blacksquare$

Remark 4.11 Definition 2.10 provided another embedding, which represents X by its ***standard resolution***,

$$\begin{tikzcd} X \arrow[r, shift left=1, "\eta_X"] & \Sigma^2 X \arrow[l, shift left=1, "\widehat{\eta_{\Sigma X}}"] \arrow[r, shift left=2, "\eta_{\Sigma^2 X}"] \arrow[r, "\Sigma^2\eta_X" description] & \Sigma^4 X. \arrow[l, shift left=2, "\widehat{\eta_{\Sigma^3 X}}"] \end{tikzcd}$$

This makes

$$\begin{tikzcd} X \arrow[r, shift left, "\eta_X"] \arrow[r, phantom, "\cong"] & \{\Sigma^2 X \mid E\}, \arrow[l, shift left, "\widehat{\eta_{\Sigma X}}"] \end{tikzcd}$$

where $E \;=\; \eta_{\Sigma^3 X} \,;\, \Sigma^3\eta_X \;=\; \Sigma\eta_X \,;\, \eta_{\Sigma^X}$ by $\mathcal{F} \mapsto \lambda F.\, F\big(\lambda x.\, \mathcal{F}(\lambda\phi.\, \phi x)\big)$. ■

The nucleus E is enough on its own to represent the object.

Lemma 4.12 For any object $(Y \rightrightarrows Z)$, we have the isomorphism in $\overline{\mathcal{C}}$

$$\begin{tikzcd} Z \arrow[d, "\widehat{J}"', shift right=3] \arrow[d, phantom, "u"] \\ Y \arrow[u, shift left=0] \arrow[u, bend right, "v"'] \end{tikzcd} \qquad \cong \qquad \begin{tikzcd} \Sigma^2 Y \arrow[d, "\widehat{\eta_{\Sigma^Y}}"', shift right=3] \arrow[d, phantom, "\eta_Y"] \\ Y \arrow[u] \arrow[u, bend right, "(\eta_Y \,;\, \Sigma^E)"'] \end{tikzcd}$$

encoded by $\widehat{E} = v \,;\, \widehat{J}$ in both directions.

Proof. The object on the right gives rise to the same nucleus because

$$\eta_Y \,;\, \Sigma^E \,;\, \widehat{\eta_{\Sigma^Y}} \;=\; \eta_Y \,;\, \widehat{\eta_{\Sigma^Y}} \,;\, \widehat{E} \;=\; \widehat{E}$$

by naturality and the unit law. It satisfies Definition 4.2 because $\eta_Y \,;\, \widehat{\eta_{\Sigma^Y}} = \mathsf{id}_Y$ and

$$\widehat{E} \,;\, \eta_Y \,;\, \Sigma^E \;=\; \widehat{E} \,;\, \eta_Y$$

since E is a nucleus. Hence both objects are $\{Y \mid E\}$, on which $\widehat{E}$ is the identity, and it also serves as the isomorphism in both directions. ■

Corollary 4.13
(a) Every Σ-split mono arises in this way.
(b) In particular, so does any first class retract.
(c) Composites of Σ-split monos are Σ-split monos (Example 2.5(c)).

Proof. Given $X \underset{\widehat{I}}{\overset{i}{\rightleftarrows}} Y$ with $i \,;\, \widehat{I} = \mathsf{id}_X$, put $\widehat{E} = \widehat{I} \,;\, i$ as in Remark 2.9.

This satisfies the $\widehat{E}$ equation because

$$\widehat{I} \,;\, i \,;\, \eta_Y \,;\, \Sigma^I \,;\, \Sigma^2 i \;=\; \widehat{I} \,;\, \eta_X \,;\, \Sigma^2 i \,;\, \Sigma^I \,;\, \Sigma^2 i \;=\; \widehat{I} \,;\, \eta_X \,;\, \Sigma^2 i \;=\; \widehat{I} \,;\, i \,;\, \eta_Y$$

by naturality of η and $i \,;\, \widehat{I} = \mathsf{id}_X$. For composition, $\widehat{E} = i_1 \,;\, i_2 \,;\, \widehat{I}_2 \,;\, \widehat{I}_1$. ■

Although $\widehat{E}$ suffices to describe an object of the new category, and will be used in the new calculus in Section 8, it is not very illuminating. The reason for introducing the Σ-split pair $(Y \rightrightarrows Z)$ is that the new object is its formal equaliser, as in Proposition 3.8.

Proposition 4.14 Every object $(Y \rightrightarrows Z)$ of $\overline{\mathcal{C}}$ is the equaliser in $\overline{\mathcal{C}}$ of the diagram that it suggests, considered to consist of images (*via* Lemma 4.10) of objects and maps in $\mathcal{C}$. If this diagram already has an equaliser X in $\mathcal{C}$, and $\Sigma^{(-)}$ takes it to a coequaliser in $\mathcal{C}$, then $(Y \rightrightarrows Z) \cong X$ in $\overline{\mathcal{C}}$.

Proof. For any object $\Gamma = (Y_0 \rightrightarrows Z_0)$ of $\overline{\mathcal{C}}$, we check that an $\mathsf{H}\mathcal{C}$-map $\widehat{H} : Y_0 \longrightarrow\!\!\!\!\times\; Y$ satisfies the conditions for being a $\overline{\mathcal{C}}$-map $\Gamma \to (Y \rightrightarrows Z)$ iff it satisfies those for being a $\overline{\mathcal{C}}$-map $\Gamma \to (Y \rightrightarrows Y)$ that has equal composites with u and v. This is simply a matter of changing the status of the equation $\widehat{H}\,;u = \widehat{H}\,;v$ from being part of Definition 4.5 of a $\overline{\mathcal{C}}$-morphism to being a test for the equaliser.

$$\begin{tikzcd}
X \arrow[r, tail, "i"] & Y \arrow[r, tail, shift left, "u"] \arrow[r, shift right, "v"'] & Z \\
\Sigma^X \arrow[r, dotted, shift right, "I"'] & \Sigma^Y \arrow[l, two heads, shift right, "\Sigma^i"'] \arrow[r, "J"] & \Sigma^Z \arrow[l, two heads, shift right=2, "\Sigma^u"'] \arrow[l, shift left=2, "\Sigma^v"]
\end{tikzcd}$$

For the second part, $J\,;\Sigma^v : \Sigma^Y \to \Sigma^Y$ has equal composites with Σ^u and Σ^v, whose coequaliser is Σ^X by hypothesis, so $E = J\,;\Sigma^v = \Sigma^i\,;I$ for some I as shown. Both $I\,;\Sigma^i$ and id mediate from the equaliser to Σ^X, so are equal by uniqueness. Then $i : X \to (Y \rightrightarrows Z)$ is an isomorphism, with inverse $\widehat{I} : (Y \rightrightarrows Z) \to X$. ■

As the construction of $\overline{\mathcal{C}}$ is one that "freely adjoins equalisers" we would expect at this point to have to show that $\overline{\mathcal{C}}$ does in fact *have* such equalisers, including those for newly defined parallel pairs, *and that* $\Sigma^{(-)}$ *takes them to coequalisers*. However, in our case, powers of Σ are involved in this statement, whereas we still have a lot of work to do to construct such powers in $\overline{\mathcal{C}}$. This will be done in Proposition 6.10.

We draw one easy corollary from the Proposition here, largely to show the contrast with the more difficult study of products that follows in the next section.

Proposition 4.15 The functor $\mathcal{C} \to \overline{\mathcal{C}}$ preserves such colimits as exist.

PROOF. Let (U_i) be a diagram in $\mathcal{C}$. Any cocone under it with vertex $(Y \rightrightarrows Z)$ in $\overline{\mathcal{C}}$ is, by the previous result, a cocone with vertex Y such that for each i the composites $U_i \to Y \rightrightarrows Z$ are equal. But this cocone might as well be in $\mathcal{C}$, and has a mediator $U = \mathsf{colim}U_i \to Y$, which also has equal composites $U \to Y \rightrightarrows Z$, and is therefore a mediator $U \to (Y \rightrightarrows Z)$ in $\overline{\mathcal{C}}$. ■

5 Injectives and products in the new category

We know from Section A 6, and from the work on continuations by others cited there, that binary products are the crucial issue. These studies have also taught us that we must take just one step at a time. In Section 7 below we sketch the construction of products in $\overline{\mathcal{C}}$ as coproducts of algebras, but the motivation of that approach involves the anticipation of results that ought to follow from the conclusion itself.

In this section we argue forwards from the structure in the given category $\mathcal{C}$, using basic (albeit pedestrian) categorical ideas, and sticking to the *objects* rather than their *algebras* of predicates. The plan is to regard $\overline{\mathcal{C}}$-objects as equalisers in $\mathcal{C}$ as in Proposition 4.14, and then use the given products in $\mathcal{C}$.

However, the universal property of these given products is only tested by diagrams of the form $\Gamma \to A \times B$ with $\Gamma \in \mathsf{ob}\mathcal{C}$, so we need to generalise this to $\Gamma \in \mathsf{ob}\overline{\mathcal{C}}$. This is more difficult than the situation for colimits in Proposition 4.15: in this case we need to be able to turn maps $\{Y \mid E\} \to A$ into $Y \to A$, which is just the property of injectivity from which we began in Remark 2.1.

Definition 5.1 An object A of $\overline{\mathcal{C}}$ is said to be ***injective*** if, for any subspace inclusion $\{Y \mid E\} \rightarrowtail Y$ and map $f : \{Y \mid E\} \to A$, there is some (not necessarily unique) map $h : Y \to A$ such that

$f = \widehat{E}\,;h$.

$$\begin{tikzcd}
\{Y \mid E\} \arrow[rr, tail, "\widehat{E}"] \arrow[dr, "f"'] & & Y \arrow[dl, dotted, "h"] \\
& A &
\end{tikzcd}$$

Proposition 5.2 An object A of $\overline{\mathcal{C}}$ is injective iff it is a retract of some Σ^U (by which we still mean the exponential in $\mathcal{C}$). Indeed, when $A \in \mathsf{ob}\mathcal{C}$ is injective, there is a map $\alpha : \Sigma^2 A \to A$ such that $\eta_A\,;\alpha = \mathsf{id}$, though this need not satisfy the other equation that is required for (A, α) to be an Eilenberg–Moore algebra, namely $\mu_A\,;\alpha = \Sigma^2\alpha\,;\alpha$.

Proof. ($\Leftarrow$) Suppose first that $A = \Sigma^U$, and put $\alpha = \Sigma^{\eta_U}$. By Definition 4.5, any morphism $\widehat{H} : \{Y \mid E\} \to A$ in $\overline{\mathcal{C}}$ satisfies

$$\widehat{H} \;=\; \widehat{H}\,;\eta_A\,;\alpha \;=\; \widehat{E}\,;\eta_Y\,;\Sigma^H\,;\alpha \;=\; \widehat{E}\,;h,$$

where $h = \eta_Y\,;\Sigma^H\,;\alpha$ is in $\mathcal{C}$. The result extends to retracts of Σ^U by composition with the inclusion and surjection.

$$\begin{tikzcd}
\{Y \mid E\} \arrow[r, tail, "\widehat{E}"] \arrow[dr, "\widehat{H} = \mathsf{id}"'] & Y \arrow[r, tail, "\eta_Y"] & \Sigma^{\Sigma^Y} \arrow[dl, dotted, "h"] \\
& A = \{Y \mid E\} &
\end{tikzcd}
\qquad
\begin{tikzcd}
\{Y \mid E\} \arrow[r, tail, "\widehat{E}"] \arrow[d, "\widehat{H}"'] & Y \arrow[d, dotted, "h"] \\
A = \{\Sigma^U \mid \Sigma^e\} \arrow[r, tail, shift left, "e"] & \Sigma^U \arrow[l, two heads, shift left, "e"]
\end{tikzcd}$$

($\Rightarrow$) The composite $\{Y \mid E\} \rightarrowtail \Sigma^2 Y$ is a subspace by Corollary 4.13; injectivity says that this is split by h. When $A = Y \in \mathsf{ob}\mathcal{C}$, $\alpha = h$ satisfies $\eta_A\,;\alpha = \mathsf{id}$. ∎

Corollary 5.3 $\overline{\mathcal{C}}$ has ***enough injectives***: each object $\{Y \mid E\}$ is a subspace of some injective $\Sigma^2 Y$. ∎

Examples 5.4 In **LKSp**, the injectives are the continuous lattices equipped with the Scott topology [Sco72], whilst the algebras are also distributive. In **Set**, injectives are (sets that carry the structure of) complete lattices and algebras are powersets (which, classically, are complete atomic Boolean algebras). ∎

Lemma 5.5 Any finite product of injectives (or algebras) is again injective (respectively, an algebra).

Proof. For the terminal object, $\alpha = !_{\Sigma\Sigma}$.

For injectives (A, α) and (B, β), we define $P_0 = \Sigma^2\pi_0\,;\alpha : \Sigma^2(A \times B) \to A$ and $P_1 = \Sigma^2\pi_1\,;\beta : \Sigma^2(A \times B) \to B$, and then $\eta_{A\times B}\,;\langle P_0, P_1\rangle = \mathsf{id}$. This follows from naturality of η, as illustrated in the right-hand trapezium below.

Moreover, if (A, α) and (B, β) are algebras then

$$\begin{aligned}
&\Sigma^2\langle P_0, P_1\rangle\,;\langle P_0, P_1\rangle\,;\pi_0 \\
&\qquad = \Sigma^4\pi_0\,;\Sigma^2\alpha\,;\alpha \\
&\qquad = \Sigma^4\pi_0\,;\Sigma\eta\Sigma A\,;\alpha \\
&\qquad = \Sigma\eta\Sigma(A \times B)\,;\Sigma^2\pi_0\,;\alpha \\
&\qquad = \Sigma\eta\Sigma(A \times B)\,;\langle P_0, P_1\rangle\,;\pi_0
\end{aligned}$$

so $(A \times B, \langle P_0, P_1 \rangle)$ is also an algebra. ∎

Lemma 5.6 The functor $\mathcal{C} \to \overline{\mathcal{C}}$ preserves the terminal object.

$$\begin{tikzcd}
\{Y \mid E\} \arrow[rr, tail, "\widehat{E}"] \arrow[dr, "\widehat{H}"'] & & Y \arrow[dl, dotted, "h"] \\
& \mathbf{1} &
\end{tikzcd}$$

Proof. Since $\mathbf{1}$ is injective, $\widehat{E}\,;\,!_Y$ is the only map $\widehat{H} : \{Y \mid E\} \to \mathbf{1}$. ∎

Lemma 5.7 The functor $\mathcal{C} \to \overline{\mathcal{C}}$ preserves binary products of injectives.

$$\begin{tikzcd}
\Sigma^2 Y \arrow[r, shift left, "\Sigma^H"] \arrow[r, shift right, "\Sigma^K"'] \arrow[rr, bend left, "\Sigma^F"] & \Sigma^2(A \times B) \arrow[r, "\Sigma^2 \pi_0"] \arrow[dd, shift left, "\langle P_0, P_1 \rangle"] & \Sigma^2 A \arrow[d, shift left, "\alpha"] \\
Y \arrow[u, "\eta_Y"] \arrow[rr, "f"] \arrow[dr, dotted, "{\langle f, g \rangle, h, k}"] & & A \arrow[u, shift left, "\eta_A"] \\
\{Y \mid E\} \arrow[u, tail, "\widehat{E}"] \arrow[r, shift left, "\widehat{H}"] \arrow[r, shift right, "\widehat{K}"'] \arrow[rru, bend right, "\widehat{F}"'] & A \times B \arrow[uu, shift left, "\eta_{A \times B}"] \arrow[ur, "\pi_0"'] &
\end{tikzcd}$$

Proof. Let $\widehat{F} : \{Y \mid E\} \to A$ and $\widehat{G} : \{Y \mid E\} \to B$ in $\overline{\mathcal{C}}$. Put

$$f \;=\; \eta_Y \,;\, \Sigma^F \,;\, \alpha \qquad \text{and} \qquad g \;=\; \eta_Y \,;\, \Sigma^G \,;\, \beta,$$

so $\widehat{F} = \widehat{E}\,;f$ and $\widehat{G} = \widehat{E}\,;g$ by Proposition 5.2, and $\langle f, g \rangle$ is given by the product in $\mathcal{C}$.

Then $\widehat{H} = \widehat{E}\,;\langle f, g \rangle : \{Y \mid E\} \to A \times B$ satisfies $\widehat{H}\,;\pi_0 = \widehat{F}$ and $\widehat{H}\,;\pi_1 = \widehat{G}$.

Suppose that $\widehat{H}$ and $\widehat{K}$ both satisfy these equations. Using Lemma 5.5, put

$$h \;=\; \eta_Y \,;\, \Sigma^H \,;\, \langle P_0, P_1 \rangle \qquad \text{and} \qquad k \;=\; \eta_Y \,;\, \Sigma^K \,;\, \langle P_0, P_1 \rangle,$$

so $\widehat{H} = \widehat{E}\,;h$ and $\widehat{K} = \widehat{E}\,;k$ by Proposition 5.2.

Since $\Sigma^H \,;\Sigma^2\pi_0 = \Sigma^F = \Sigma^K \,;\Sigma^2\pi_0$, we have $h\,;\pi_0 = f = k\,;\pi_0$ by the way in which h, k and f were constructed, and similarly $h\,;\pi_1 = g = k\,;\pi_1$. Hence $h = k = \langle f, g \rangle$, as these maps are in $\mathcal{C}$, so $\widehat{H} = \widehat{K}$. ∎

Now we can begin the main business of this section. In view of Corollary 5.3, we may assume that the $\overline{\mathcal{C}}$-objects of which we want to form the product are given by Σ-split pairs of maps *between injectives*. In fact this restriction will become redundant, as the argument below can be used first to show that *all* products are preserved, and then to construct products of non-injective pairs.

Lemma 5.8 Let $(Y \rightrightarrows Z) \in \mathsf{ob}\overline{\mathcal{C}}$ and $W \in \mathsf{ob}\mathcal{C}$. Then

$$\begin{tikzcd} Y \times W \arrow[rr, bend left, "u \times W"] \arrow[rr, bend right, "v \times W"'] & \widehat{J} \times W \arrow[l, tail] \arrow[r, no head] & Z \times W \end{tikzcd}$$

is also an object of $\overline{\mathcal{C}}$. It is called $(Y \rightrightarrows Z) \times W$ because, at least when U, V and W are injective, this is the product in $\overline{\mathcal{C}}$. In the comprehension notation,

$$\{Y \mid E\} \times W \;=\; \{Y \times W \mid E^W\} \xrightarrow{\widehat{E^W}} Y \times W.$$

This is functorial with respect to

$$g : W_1 \to W_2, \quad \text{where } \{Y \mid E\} \times g = \widehat{E^g}$$

(recall that $\mathsf{id}_{\{Y|E\}} = \widehat{E}$) and to

$$\widehat{F} : \{Y_1 \mid E_1\} \to \{Y_2 \mid E_2\}, \qquad \text{where } \widehat{F} \times W = \widehat{F^W}.$$

Proof. Apply $(-)^W$ to the equations in Definition 4.2 to get $J^W \,;\Sigma^{u\times W} = \mathsf{id}$ and

$$\Sigma^{v\times W} \,; J^W \,; \Sigma^{u\times W} \;=\; \Sigma^{v\times W} \,; J^W \,; \Sigma^{v\times W}.$$

Proposition 4.14 said that the formal pairs are equalisers, which property commutes with products. In comprehension notation, $\Sigma^{v\times W} \,; J^W = E^W$.

The formula for $\{Y \mid E\} \times g$ is merely naturality of $\widehat{E^{(-)}}$.

Using the injectivity of Y_2, we may write $\widehat{F}$ as

$$\begin{tikzcd} Y_1 \arrow[r, dotted, "f"] & Y_2 \\ \{Y_1 \mid E_1\} \arrow[u, "\widehat{E_1}"] \arrow[r, "\widehat{F}"] & \{Y_2 \mid E_2\} \arrow[u, "\widehat{E_2}"'] \end{tikzcd}$$

for some (not necessarily unique) f, where (by Definition 4.5) $\widehat{F} = \widehat{F} \,; \widehat{E_2} = \widehat{E_1} \,; f$, so $F = \Sigma^f \,; E_1$. Applying $(-) \times W$ to the commutative square, the formula for $\widehat{F} \times W$ has to be $\widehat{E_1} \,; (f \times W) = \widehat{G}$ where $G \equiv (\Sigma^f \,; E_1)^W = F^W$. ■

Remark 5.9 Alternatively, one could show directly that $\widehat{E^g}$ and $\widehat{F^W}$ are morphisms, and that they satisfy the equations for the product functor. For $\widehat{F^W}$, we need to use the ***strength***

$$\begin{tikzcd} (\Sigma^2 Y) \times W \arrow[rr, "\sigma_{Y,W} \,:\, G, w \mapsto \lambda\theta.\, G\left(\lambda y.\, \theta(y, w)\right)"] & & \Sigma^2(Y \times W) \\ & Y \times W \arrow[ul, "\eta_Y \times W"] \arrow[ur, "\eta_{Y\times W}"'] & \end{tikzcd}$$

for the monad, which is a natural transformation and obeys the equation shown.

Remark 5.10 The product of two Σ-split pairs must form a pullback (intersection) in $\overline{\mathcal{C}}$, in the same way that the product of *any* two maps in a category gives rise to a pullback.

$$
\begin{tikzcd}
 & X_1 \times X_2 \arrow[r, dotted] \arrow[d, dotted] \arrow[dr, phantom, "\lrcorner", very near start] & X_1 \times Y_2 = \{Y_1 \times Y_2 \mid E_1^{Y_2}\} \arrow[d] \\
\{Y_1 \times Y_2 \mid E_2^{Y_1}\} = & Y_1 \times X_2 \arrow[r, tail] & Y_1 \times Y_2
\end{tikzcd}
$$

Unfortunately, our "comprehension" notation does not admit intersections — either abstractly or in the leading example of **LKLoc** — so we still have to find an E on $Y_1 \times Y_2$ to describe the subspace. If $E_1^{Y_2}$ and $E_2^{Y_1}$ commute (as they do if $E_1 = \Sigma^{e_1}$ or $E_2 = \Sigma^{e_2}$) then their composite encodes the intersection. But it turns out that, even if $E_1^{Y_2} ; E_2^{Y_1} \neq E_2^{Y_1} ; E_1^{Y_2}$, *either* of them will do the job: of course products and pullbacks are unique up to isomorphism, but we have two representations of them.

Lemma 5.11 The composite

$$
\begin{tikzcd}[column sep=huge]
Y_1 \times Y_2 \arrow[r, shift left=6, tail, "u_1 \times Y_2"] \arrow[r, "v_1 \times Y_2" description] & Z_1 \times Y_2 \arrow[l, shift left=6, "\widehat{J_1} \times Y_2"] \arrow[r, shift left=6, tail, "Z_1 \times u_2"] \arrow[r, "Z_1 \times v_2" description] & Z_1 \times Z_2 \arrow[l, shift left=6, "Z_1 \times \widehat{J_2}"]
\end{tikzcd}
$$

is an object of $\overline{\mathcal{C}}$, where

$$
\begin{array}{rcl}
u & = & u_1 \times Y_2 ; Z_1 \times u_2 \;=\; Y_1 \times u_2 ; u_1 \times Z_2 \;=\; u_1 \times u_2 \\
v & = & v_1 \times Y_2 ; Z_1 \times v_2 \;=\; Y_1 \times v_2 ; v_1 \times Z_2 \;=\; v_1 \times v_2 \\
\widehat{J} & = & Z_1 \times \widehat{J_2} ; \widehat{J_1} \times Y_2 \;\neq\; \widehat{J_1} \times Z_2 ; Y_1 \times \widehat{J_2} \\
\widehat{E} & = & v ; \widehat{J} \;=\; Y_1 \times \widehat{E_2} ; \widehat{E_1} \times Y_2 \;\subset_{Y_1 \times Y_2}\; Y_1 \times \widehat{E_2}
\end{array}
$$

the inclusion on the last line being in the sense of Notation 4.4.

PROOF. Clearly $u ; \widehat{J} = \mathsf{id}$, whilst

$$
\begin{array}{rcll}
\widehat{E} \equiv v ; \widehat{J} & = & v_1 \times Y_2 ; Z_1 \times v_2 ; Z_1 \times \widehat{J_2} ; \widehat{J_1} \times Y_2 & \text{definitions of } v \text{ and } J \\
 & = & v_1 \times Y_2 ; Z_1 \times \widehat{E_2} ; \widehat{J_1} \times Y_2 & \text{definition of } E_2 \\
 & = & Y_1 \times \widehat{E_2} ; v_1 \times Y_2 ; \widehat{J_1} \times Y_2 & v_1 \text{ central} \\
 & = & Y_1 \times \widehat{E_2} ; \widehat{E_1} \times Y_2. & \text{definition of } E_1.
\end{array}
$$

Using the given equations of the form $v_i ; \widehat{J_i} ; u_i = v_i ; \widehat{J_i} ; v_i$, we must prove this equation for the new Σ-split pair:

$$
\begin{array}{rcl}
v ; \widehat{J} ; u & = & Y_1 \times \widehat{E_2} ; \widehat{E_1} \times Y_2 ; v_1 \times Y_2 ; Z_1 \times v_2 \\
 & = & Y_1 \times (v_2 ; \widehat{J_2} ; u_2) ; (v_1 ; \widehat{J_1} ; u_1) \times Z_2 \\
 & = & v ; \widehat{J} ; v,
\end{array}
$$

using centrality of u_2, and the same argument with v in place of u.

$$
\begin{tikzcd}
Y_1 \times Y_2 \arrow[r, "Y_1 \times \widehat{E_2}"] & Y_1 \times Y_2 \arrow[r, "\widehat{E_1} \times Y_2"] \arrow[d, shift right, "Y_1 \times v_2"'] \arrow[d, shift left, tail, "Y_1 \times u_2"] & Y_1 \times Y_2 \arrow[d, shift right, "Y_1 \times v_2"'] \arrow[d, shift left, tail, "Y_1 \times u_2"] \arrow[dr, "\mathsf{id}"] & \\
 & Y_1 \times Z_2 \arrow[r, "\widehat{E_1} \times Z_2"] & Y_1 \times Z_2 \arrow[r, "Y_1 \times \widehat{J_2}"] & Y_1 \times Y_2
\end{tikzcd}
$$

Finally, $Y_1 \times \widehat{E_2} ; \widehat{E_1} \times Y_2 ; Y_1 \times \widehat{E_2} \;=\; Y_1 \times \widehat{E_2} ; \widehat{E_1} \times Y_2$ by centrality of u_2 and v_2 with respect to $\widehat{E_1}$, and $u_2 ; \widehat{J_2} = \mathsf{id}$. ■

Proposition 5.12 The category $\overline{\mathcal{C}}$ has finite products, where

$$\{Y_1 \mid E_1\} \times \{Y_2 \mid E_2\} \;=\; \{Y_1 \times Y_2 \mid E_1^{Y_2} ; E_2^{Y_1}\},$$

whilst the product of the morphisms $\widehat{F_1} : X_1' \to X_1$ and $\widehat{F_2} : X_2' \to X_2$ is $\widehat{F_1^{Y_2'}} ; \widehat{F_2^{Y_1}}$.

Moreover, the functor $\mathcal{C} \to \overline{\mathcal{C}}$ takes products in $\mathcal{C}$ to products in $\overline{\mathcal{C}}$.

Proof. First consider the product of two objects $X_1, X_2 \in \mathsf{ob}\mathcal{C}$, expressed in $\overline{\mathcal{C}}$ by means of their standard resolutions, so the products $\Sigma^2 X_1 \times \Sigma^2 X_2$ *etc.* are preserved (Lemma 5.7). The formal product of the Σ-split pairs (Lemma 5.11) provides an equaliser in $\overline{\mathcal{C}}$ (Proposition 4.14), which is the required product, isomorphic to $X_1 \times X_2$, *i.e.* this is preserved.

The construction is then applicable to any two Σ-split pairs, so all binary products exist in $\overline{\mathcal{C}}$. Functoriality follows from that in Lemma 5.8. ■

Remark 5.13 The abstract situation of idempotents $\epsilon = (E_2)^{Y_1}$ and $\epsilon' = (E_1)^{Y_2}$ (in this case on the object $\Sigma^{Y_1 \times Y_2}$) such that $\epsilon ; \epsilon'$ and $\epsilon' ; \epsilon$ are also idempotent is studied in [D, Lemma 7.1]. It is shown there that the splittings of $\epsilon ; \epsilon'$ and $\epsilon' ; \epsilon$ are isomorphic and form a pushout diagram, in our case of the algebras $\Sigma^{X_1 \times X_2}$ *etc.* corresponding to objects in the above pullback.

6 Structure in the new category

Now that we have overcome the main difficulty concerning binary products, we are equipped to tackle exponentials, and then to show that $\overline{\mathcal{C}}$ does in fact have the monadic property for which it was designed. We still rely on injectivity, first re-casting Proposition 5.2 in terms of the double transpose of h.

Lemma 6.1 There is a natural bijection between $\overline{\mathcal{C}}$-maps $\widehat{H} : \{Y \mid E\} \to \Sigma^U$ and $\mathcal{C}$-maps $p : U \to \Sigma^Y$ such that $p = p ; E$.

Proof. Let $\widehat{H} \mapsto p = \eta_U ; H$, which satisfies $p = p ; E$ by Lemma 4.6. Conversely, let $p \mapsto \widehat{H} = \widehat{E} ; \eta_Y ; \Sigma^p$, which is a composite of $\overline{\mathcal{C}}$-maps $\{Y \mid E\} \rightarrowtail Y \longrightarrow \Sigma^U$.

Then $\widehat{H} \mapsto \widehat{E} ; \eta_Y ; \Sigma^p = \widehat{E} ; \eta_Y ; \Sigma^H ; \Sigma^{\eta_U} = \widehat{H} ; \eta_{\Sigma^U} ; \Sigma^{\eta_U} = \widehat{H}$ by Definition 4.5 and the unit law. Conversely, $p \mapsto \eta_U ; H = \eta_U ; \Sigma^2 p ; \Sigma^{\eta_Y} ; E = p ; \eta_{\Sigma^Y} ; \Sigma^{\eta_Y} ; E = p ; E = p$ by naturality for η and the unit law.

For naturality, let $\widehat{F} : \{Y' \mid E'\} \to \{Y \mid E\}$ and $g : U' \to U$. Then p becomes $g ; p ; F$. ■

Proposition 6.2 $\overline{\mathcal{C}}$ has and $\mathcal{C} \to \overline{\mathcal{C}}$ preserves powers of Σ, where

$$\Sigma^{\{Y \mid E\}} \;=\; \{\Sigma^Y \mid \Sigma^E\} \quad \text{and} \quad \Sigma^{\widehat{F}} = F.$$

$$\begin{tikzcd}
Y_1 \times Y_2 \arrow[r, "h"] & \Sigma \\
X_1 \times X_2 \arrow[u, "\widehat{E_1^{Y_2}} ; \widehat{E_2^{Y_1}}"] \arrow[ur, "\widehat{H}"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
Y_1 \arrow[r, "k"] & \Sigma^{Y_2} \arrow[d, shift left, "E_2"] \\
X_1 \arrow[u, "\widehat{E_1}"] \arrow[r, "\widehat{K}"] & \Sigma^{X_2} \arrow[u, shift left, "E_2"]
\end{tikzcd}$$

Proof. $\overline{\mathcal{C}}$-maps $\widehat{H} : X_1 \times X_2 \to \Sigma$ correspond bijectively to $p : \mathbf{1} \to \Sigma^{Y_1 \times Y_2}$ in $\mathcal{C}$ such that $p = p ; E_1^{Y_2} ; E_2^{Y_1}$, by Lemma 6.1. By Remark 5.10, this is equivalent to $p ; E_1^{Y_2} = p = p ; E_2^{Y_1}$. Let $k : Y_1 \to \Sigma^{Y_2}$ and $q : Y_2 \to \Sigma^{Y_1}$ be the exponential transposes of p, which satisfy $k = k ; E_2$ and

$q = q \,; E_1$. But then q corresponds to $\widehat{K} : X_1 \to \Sigma^{Y_2}$ by Lemma 6.1, and $\widehat{K} = \widehat{K} \,; E_2$ by the other equation, so it factors through Σ^{X_2}. Conversely, $\widehat{K}$ defines k, q, p and $\widehat{H}$ in the same way.

Now let $\widehat{F_1} : X_1' \to X_1$ and $\widehat{F_2} : X_2' \to X_2$, whose product is $\widehat{F_1^{X_2'}} \,; \widehat{F_2^{X_1}}$ by Proposition 5.12. Then by the naturality part of the Lemma, p becomes $p \,; F_2^{X_1} \,; F_1^{X_2'}$ and $\widehat{K}$ becomes $\widehat{F_1} \,; \widehat{K} \,; F_2$. ■

Lemma 6.3 $\eta_X : X \to \Sigma^2 X$ in $\overline{\mathcal{C}}$ is given by $\widehat{E} \,; \eta_Y : Y \longrightarrow\!\!\!\!\!\times\, \Sigma^2 Y$.

$$\begin{tikzcd}
Y \arrow[rr, "\eta_Y"] & & \Sigma^2 Y \arrow[d, shift left, "\Sigma^E"] \\
X = \{Y \mid E\} \arrow[u, "\widehat{E}"] \arrow[rr, "\eta_X"] & & \Sigma^2 X \arrow[u, shift left, "\Sigma^E"]
\end{tikzcd}$$

PROOF. From functoriality in the Proposition, Σ^2 takes the inclusion $\widehat{E}$ to Σ^E. Then for η to be natural on $\overline{\mathcal{C}}$, we need $\eta_X = \widehat{E} \,; \eta_Y \,; \Sigma^E$. But this expression is the left hand side of the defining equation for the nucleus E (Definition 4.3), the right hand side being $\widehat{E} \,; \eta_Y$.

Alternatively, we may see $\eta_{\{Y|E\}}$ as the double transpose of $\mathsf{id}_{\{Y|E\}} = \widehat{E}$. This correspondence is that between k and q on the right hand side of the Proposition, in the case of $X_1 = \Sigma^X$, $X_2 = X$. Then $\widehat{K} = \mathsf{id}_{\Sigma^X} = E = k$ and $q = \eta_Y \,; \Sigma^E$, so $\eta_{\{Y|E\}} = \widehat{E} \,; q = \widehat{E} \,; \eta_Y$. ■

Corollary 6.4 $\mathbf{Alg}_{\overline{\mathcal{C}}} \simeq \mathbf{Alg}_{\mathsf{KS}\mathcal{C}} \cong \mathbf{Alg}_{\mathsf{K}\mathcal{C}}$, *cf.* $\mathbf{Alg}_{\mathsf{S}\mathcal{C}} \cong \mathbf{Alg}_{\mathcal{C}}$ in Lemma A 7.3.

PROOF. Let (A, α) be an algebra over $\overline{\mathcal{C}}$, with $A = \{Y \mid E\}$. Then $A \lhd \Sigma^2 A \lhd \Sigma^2 Y$, so A may as well be in $\mathsf{K}\mathcal{C}$, since this is fully embedded in $\overline{\mathcal{C}}$ by Corollary 4.13. ■

Lemma 6.5 $\mathsf{H}\overline{\mathcal{C}}$-maps $\widehat{H} : X_1 \longrightarrow\!\!\!\!\!\times\, X_2$ are given by

$$H : \Sigma^{Y_2} \to \Sigma^{Y_1} \quad \text{such that} \quad H \,; E_1 \;=\; H \;=\; E_2 \,; H.$$

So $\mathsf{H}\overline{\mathcal{C}} \subset \mathsf{KH}\mathcal{C}$ is the full subcategory consisting of those objects $(X, \widehat{E})$ for which E is a nucleus.

$$\begin{tikzcd}
\Sigma^{Y_2} \arrow[rr, "H"] \arrow[d, shift left, "E_2"] & & \Sigma^{Y_1} \arrow[d, shift left, "E_1"] \\
\Sigma^{X_2} \arrow[u, shift left, "E_2"] \arrow[rr] & & \Sigma^{X_1} \arrow[u, shift left, "E_1"]
\end{tikzcd}$$

PROOF. An $\mathsf{H}\overline{\mathcal{C}}$-map $\widehat{H} : X_1 \longrightarrow\!\!\!\!\!\times\, X_2$ is a $\overline{\mathcal{C}}$-map $H : \Sigma^{X_2} \to \Sigma^{X_1}$. We have just shown that these exponentials are retracts, so this map is $H : \Sigma^{Y_2} \to \Sigma^{Y_1}$ (now just in $\mathcal{C}$) such that $E_2 \,; H = H = H \,; E_1$. This is also how morphisms of $\mathsf{KH}\mathcal{C}$ are defined (Remark 4.1). ■

Lemma 6.6 An $\mathsf{H}\overline{\mathcal{C}}$-map $\widehat{H} : X_1 \longrightarrow\!\!\!\!\!\times\, X_2$ is first class (a $\overline{\mathcal{C}}$-map) iff it respects naturality of η, *i.e.* $\eta_{X_1} \,; \Sigma^2 \widehat{H} = \widehat{H} \,; \eta_{X_2}$, or $H : \Sigma^{X_2} \to \Sigma^{X_1}$ is a homomorphism.

PROOF. Using the previous two lemmas,

$$\eta_{X_1} \,; \Sigma^2 \widehat{H} \;=\; \widehat{E_1} \,; \eta_{Y_1} \,; \Sigma^2 \widehat{H} \quad \text{and} \quad \widehat{H} \,; \eta_{X_2} \;=\; \widehat{H} \,; \widehat{E_2} \,; \eta_{Y_2} \;=\; \widehat{H} \,; \eta_{Y_2}.$$

Equality of these is the second equation in Definition 4.5. ■

Corollary 6.7 Every homomorphism $\widehat{K} : \Sigma^{X_2} \to \Sigma^{X_1}$ in $\overline{\mathcal{C}}$ is of the form $\widehat{K} = \Sigma^{\widehat{H}}$ for some unique $\overline{\mathcal{C}}$-map $\widehat{H} : X_1 \to X_2$. From Section A 7 we deduce that all objects of $\overline{\mathcal{C}}$ are sober, that $\mathsf{S}\overline{\mathcal{C}} \cong \overline{\mathcal{C}}$ and that $X \times (-)$ distributes over such colimits as exist in $\overline{\mathcal{C}}$. ■

Notation 6.8 Recall from Definition A 6.3 that we wrote $\mathsf{force} : \Sigma^2 X \longrightarrow\!\!\!\!\!\times\ X$ for the $\mathsf{H}\mathcal{C}$-map $\widehat{\eta_{\Sigma^X}}$, and that this is natural with respect to $\mathsf{H}\mathcal{C}$-maps $\widehat{H} : X \longrightarrow\!\!\!\!\!\times\ Y$. Similarly, we now write

$$\mathsf{admit} \quad \text{for} \quad \widehat{I} : Y \longrightarrow\!\!\!\!\!\times\ \{Y \mid E\}.$$

Like the identity, the inclusion $i : (Y \rightrightarrows Z) \hookrightarrow Y$ and its splitting $\widehat{I}$ are both encoded as $\widehat{E}$. In Section 8 we shall add an operator admit to the λ-calculus, just as we introduced focus in Section A 8, and they both have side-conditions. However, we shall not on this occasion give different names to the operator in $\overline{\mathcal{C}}$ and the map in $\mathsf{H}\mathcal{C}$.

Lemma 6.9 $\mathsf{force}_{\{Y|E\}} = \Sigma^2 i \,;\, \mathsf{force}_Y \,;\, \mathsf{admit}$.

$$\begin{tikzcd}[column sep=huge, row sep=huge]
\Sigma^2\{Y \mid E\} \arrow[r, shift left, tail, "\Sigma^2 i"] \arrow[d, shift left, "\mathsf{force}_{\{Y|E\}}"] & \Sigma^2 Y \arrow[l, shift left, two heads, "\Sigma^2\,\mathsf{admit}"] \arrow[d, shift left, "\mathsf{force}_Y"] \\
\{Y \mid E\} \arrow[u, shift left, "\eta_{\{Y|E\}}"] \arrow[r, shift left, tail, "i"] & Y \arrow[u, shift left, "\eta_Y"] \arrow[l, shift left, "\mathsf{admit}"]
\end{tikzcd}$$

Hence if $\Gamma \vdash P : \Sigma^2\{Y \mid E\}$ is prime then so is $\Gamma \vdash \Sigma^2 i\, P : \Sigma^2 Y$ (Lemma A 4.6) and

$$\mathsf{focus}_{\{Y|E\}}\, P \;=\; \mathsf{admit}\big(\mathsf{focus}_Y(\Sigma^2 i\, P)\big).$$

Proof. The diagram shows naturality of force with respect to admit in $\mathsf{H}\overline{\mathcal{C}}$, *i.e.* of η with respect to E in $\mathcal{C}$. Recall from Lemma A 4.3 that P is prime iff it has equal composites with $\Sigma^2\eta_{(-)}$ and $\eta\Sigma^2(-)$, which are natural with respect to $i : \{Y \mid E\} \rightarrowtail Y$. ■

Now we can show that $\overline{\mathcal{C}}$ has the new structure that it was introduced to provide.

Proposition 6.10 Σ-split equalisers exist in $\overline{\mathcal{C}}$, and $\Sigma^{(-)}$ sends them to coequalisers. Indeed

$$\{\{Y \mid E_1\} \mid E_2\} \;=\; \{Y \mid E_2\}$$

where $X = \{Y \mid E_1\}$ is a $\overline{\mathcal{C}}$-object and E_2 data on it for a $\overline{\overline{\mathcal{C}}}$-object $\{X \mid E_2\}$.

Proof. The defining equations for $\widehat{E_2}$ to be an $\mathsf{H}\overline{\mathcal{C}}$-map and to be a nucleus (Definition 4.3), *i.e.* to define an $\overline{\overline{\mathcal{C}}}$-object, are

$$\widehat{E_1}\,;\widehat{E_2} \;=\; \widehat{E_2} \;=\; \widehat{E_2}\,;\widehat{E_1} \quad \text{and} \quad \widehat{E_2}\,;\eta_X\,;\Sigma^{E_2} \;=\; \widehat{E_2}\,;\eta_X,$$

the first of which says that $\widehat{E_2} \subset \widehat{E_1}$ in the sense of Notation 4.4. The second equation, between $\mathsf{H}\overline{\mathcal{C}}$-maps, reduces to

$$\widehat{E_2}\,;\eta_Y\,;\Sigma^{E_2} \;=\; \widehat{E_2}\,;\eta_Y$$

in $\mathsf{H}\mathcal{C}$, but this is just the definition of a nucleus E_2 on Y. Hence $\{Y \mid E_2\}$ as a $\overline{\mathcal{C}}$-object, which Lemma 4.12 expresses as the equaliser of a Σ-split pair $Y \rightrightarrows \Sigma^2 Y$:

$$\begin{tikzcd}[column sep=huge]
\{Y \mid E_2\} \arrow[r, shift left, tail] & X = \{Y \mid E_1\} \arrow[l, shift left] \arrow[r, shift left, tail] & Y \arrow[l, shift left] \arrow[r, shift left=2] \arrow[r] & \Sigma^2 Y \arrow[l, shift left=2]
\end{tikzcd}$$

Finally, $\Sigma^{(-)}$ acts on this diagram in $\overline{\mathcal{C}}$ just as it does in $\mathcal{C}$, taking it to a split coequaliser, as in Proposition 4.14. ∎

Corollary 6.11 $(\overline{\mathcal{C}}, \Sigma)$ is monadic and $\overline{\mathcal{C}} \simeq \mathbf{Alg}^{\mathsf{op}}_{\mathsf{K}\mathcal{C}}$.

Proof. Recall that being monadic means that $\overline{\mathcal{C}} \simeq \mathbf{Alg}^{\mathsf{op}}_{\overline{\mathcal{C}}}$. By Theorem 3.3, this happens iff all objects are sober (which they are by Corollary 6.7) and all algebras are spatial. The latter follows from the previous result by Proposition 3.8. Finally, the equivalence with $\mathbf{Alg}^{\mathsf{op}}_{\mathsf{K}\mathcal{C}}$ is given by Corollary 6.4. ∎

Theorem 6.12 $(\overline{\mathcal{C}}, \Sigma)$ is the (Karoubi and) ***monadic completion*** of $(\mathcal{C}, \Sigma)$:

Let $(\mathcal{D}, \Sigma_{\mathcal{D}})$ be monadic and suppose that idempotents split in $\mathcal{D}$ (so $\mathcal{D}$ has products, powers of $\Sigma_{\mathcal{D}}$, equalisers of $\Sigma_{\mathcal{D}}$-split pairs and $\Sigma_{\mathcal{D}}^{(-)}$ sends them to coequalisers.) Let $F : \mathcal{C} \to \mathcal{D}$ be a functor that preserves products and powers of Σ. Then there is a functor $\overline{F} : \overline{\mathcal{C}} \to \mathcal{D}$ that also preserves these things and makes the square commute, and it is unique up to *equivalence*.

$$
\begin{tikzcd}
\overline{\mathcal{C}} \arrow[r, dotted, "\overline{F}"] & \overline{\mathcal{D}} \arrow[d, equal] \\
\mathcal{C} \arrow[u] \arrow[r, "F"] & \mathcal{D}
\end{tikzcd}
$$

Proof. Any object of $\overline{\mathcal{C}}$ is interpreted as a Σ-split equaliser diagram in $\mathcal{C}$ as in Proposition 4.14. Such diagrams are preserved by the functor F, but the diagram in $\mathcal{D}$ has an equaliser, which is the required image of the given $\mathcal{C}$-object. F commutes with the representation of $\times$ and $\Sigma^{(-)}$ in Propositions 5.12 and 6.2. ∎

Remark 6.13 The sober, Karoubi and monadic completions are related by the diagram

$$
\begin{tikzcd}
\mathbf{Alg}^{\mathsf{op}}_{\mathcal{C}} \arrow[r, equal] & \mathbf{Alg}^{\mathsf{op}}_{\mathsf{S}\mathcal{C}} \arrow[r, tail] & \mathbf{Alg}^{\mathsf{op}}_{\mathsf{K}\mathcal{C}} \arrow[r, equal] & \mathbf{Alg}^{\mathsf{op}}_{\overline{\mathcal{C}}} \arrow[r, equal] & \overline{\overline{\mathcal{C}}} \arrow[d, equal] \\
\mathsf{S}\mathcal{C} \arrow[u, tail] \arrow[r, tail] & \mathsf{SK}\mathcal{C} \arrow[r, tail] & \mathsf{KS}\mathcal{C} \arrow[r, equal] & \mathsf{SKS}\mathcal{C} \arrow[u, tail] \arrow[r, tail] & \overline{\mathcal{C}} \\
\mathcal{C} \arrow[u, tail] \arrow[r, tail] & \mathsf{K}\mathcal{C} \arrow[u, tail] & & &
\end{tikzcd}
$$

in which $\rightarrowtail$ denotes a full inclusion and $=\!=$ an equivalence. We leave the interested reader to find a properly general recursive idempotent on $\mathbb{N}$ (so $\mathsf{SK}\mathcal{C} \neq \mathsf{KS}\mathcal{C}$, *cf.* Remark A 9.12), and to show that any retract of a sober object is sober (so $\mathsf{KS}\mathcal{C} = \mathsf{SKS}\mathcal{C}$). ∎

The development so far has been entirely set in the abstract situation of a category $\mathcal{C}$ with an exponentiating object Σ, for which all objects are sober. In the intended applications to topology and computation, Σ is a distributive lattice and satisfies the Euclidean principle [C]; this structure is preserved by the inclusion $\mathcal{C} \to \overline{\mathcal{C}}$ since it preserves products and powers of Σ.

We also want $\mathcal{C}$ to have a natural numbers object $\mathbb{N}$, *i.e.* to admit primitive recursion at all $\mathcal{C}$-types, although sobriety of $\mathbb{N}$ means that *general* recursion is also defined (Sections A 9–10). The extension of recursion to $\overline{\mathcal{C}}$-types follows essentially the same argument as in Proposition 4.15 for colimits.

Proposition 6.14 The inclusion $\mathcal{C} \to \overline{\mathcal{C}}$ preserves $\mathbb{N}$.

PROOF. The recursion data in the fully parametrised version of this result consist of

$$z : \Gamma \to \{Y \mid E\} \quad \text{and} \quad s : \Gamma \times \mathbb{N} \times \{Y \mid E\} \to \{Y \mid E\},$$

and we shall prove it symbolically like this in Lemma 8.14. Here, for brevity, we omit the parameters in s.

First, $\{Y \mid E\}$ may be expressed as an equaliser as shown.

$$\begin{tikzcd}
 & \Gamma \arrow[dl, "0"'] \arrow[d, "z"] & & \\
\mathbb{N} \arrow[r, dotted] & \{Y \mid E\} \arrow[r, tail] & \Sigma^2 Y \arrow[r, shift left] \arrow[r, shift right] & \Sigma^4 Y \\
\mathbb{N} \arrow[u, "+1"] \arrow[r, dotted] & \{Y \mid E\} \arrow[u, "s"] \arrow[r, tail] & \Sigma^2 Y \arrow[u, "S"'] \arrow[r, shift left] \arrow[r, shift right] & \Sigma^4 Y \arrow[u]
\end{tikzcd}$$

Then use injectivity to extend $s : \{Y \mid E\} \to \{Y \mid E\}$ to endofunctions of $\Sigma^2 Y$ and $\Sigma^4 Y$, so that s becomes a mediator between equalisers. Then recursion in $\mathcal{C}$ defines a map $\mathbb{N} \to \Sigma^2 Y$ that makes the squares commute and also has equal composites to $\Sigma^4 Y$. Finally, the required map $\mathbb{N} \to \{Y \mid E\}$ mediates to the equaliser. ■

7 Algebras

A perhaps more obvious way to construct the monadic completion of $\mathcal{C}$ (as I did in 1993) is to consider the algebras, and in fact $\overline{\mathcal{C}} \simeq \mathbf{Alg}^{\mathsf{op}}_{\mathsf{K}\mathcal{C}}$ does the job. As in other approaches, the central difficulty remains that of binary products in $\overline{\mathcal{C}}$, *i.e.* coproducts of algebras. For convenience, we assume in this section alone that idempotents split in $\mathcal{C}$, so $\mathsf{K}\mathcal{C} \simeq \mathcal{C}$.

The equivalence $\overline{\mathcal{C}} \simeq \mathbf{Alg}^{\mathsf{op}}$ says that every algebra (A, α) is to be $(\Sigma^X, \Sigma^{\eta_X})$ for some new object $X \in \mathsf{ob}\overline{\mathcal{C}}$ (Definition 3.1). This means that we have to prove properties of A on the basis of its algebra structure that would be obvious if we already knew that $A = \Sigma^X$. The key such property turns out to be the double exponential transposition,

$$\mathcal{A}(\Sigma^A, B) \;=\; \overline{\mathcal{C}}(Y, \Sigma^X) \;\cong\; \overline{\mathcal{C}}(X, \Sigma^Y) \;=\; \mathcal{A}(\Sigma^B, A),$$

where X and Y are the $\overline{\mathcal{C}}$-objects corresponding to the algebras A and B, and Σ^X corresponds to Σ^A because the functor $\mathcal{C} \to \overline{\mathcal{C}}$ is supposed to preserve $\Sigma^{(-)}$.

In effect, using algebras means that we insist on representing each $\overline{\mathcal{C}}$-object in "normal form" by means of its standard resolution

$$\begin{tikzcd}[column sep=large]
X \arrow[r, tail, shift left, "\eta_X"] & \Sigma^2 X \arrow[l, shift left, "\widehat{\eta_{\Sigma^X}}"] \arrow[r, tail, shift left=2, "\eta_{\Sigma^2 X}"] \arrow[r, "\Sigma^2 \eta_X" description] & \Sigma^4 X \arrow[l, shift left=2, "\widehat{\eta_{\Sigma^3 X}}"]
\end{tikzcd}$$

$$\begin{tikzcd}[column sep=large]
A \arrow[r, shift right, "\eta_A"'] & \Sigma^2 A \arrow[l, two heads, shift right, "\alpha"'] \arrow[r, shift right=2, "\eta_{\Sigma^2 A}"'] & \Sigma^4 A, \arrow[l, two heads, shift right=2, "\mu_A = \Sigma^{\eta_{\Sigma^A}}"'] \arrow[l, "\Sigma^2 \alpha" description]
\end{tikzcd}$$

whereas Section 4 was much more flexible about Σ-split pairs. For example, Lemma 5.8 gave the obvious formula for $(Y \rightrightarrows Z) \times W$, but this does not take standard resolutions to standard resolutions: the structure map α^W that it provides must be composed with a map

$$\Sigma^{\tau_{\Sigma^X,W}} : \Sigma^{\Sigma^{\Sigma(X\times W)}} \longrightarrow \Sigma^{(\Sigma^{\Sigma^X}\times W)}.$$

When we generalise from Σ^X to A, we must replace $\Sigma^{X\times W}$ by A^W.

Remark 7.1 For any algebra (A, α) and object W, the object A^W splits the idempotent E^W, where A itself splits $E = \alpha\,;\eta_A$. Then A^W carries a ***power algebra*** structure, $\Sigma^{\tau_{A,W}}\,;\alpha^W$, where the natural transformation

$$\tau_{A,W} : W \times \Sigma^A \to \Sigma^{A^W} \qquad \text{by} \qquad x, \phi \mapsto \lambda f.\, \phi(fx)$$

plays a similar role to that of the strength σ of the monad (Remark 5.9). If $H : A \to B$ is a homomorphism them so is $H^W : A^W \to B^W$, by naturality of $\tau_{(-),W}$. ■

In fact, for any strong monad T whose algebras are exponentiable, an algebra structure $T(A^W) \to A^W$ is given by the transpose of

$$W \times T(A^W) \xrightarrow{\sigma_{W,A^W}} T(W \times A^W) \xrightarrow{T\mathsf{ev}} TA \xrightarrow{\alpha} A.$$

Remark 7.2 When X and Y are expressed by means of the standard resolution, the pullback in Remark 5.10 is as shown on the left. As $\Sigma^{X\times\Sigma^2 Y} = (\Sigma^X)^{(\Sigma^2 Y)} = A^{\Sigma^B}$, the corresponding diagram in **Alg** is the one on the right, and is to be a pushout of homomorphisms.

$$\begin{tikzcd}
X \times Y \arrow[r, "X\times\eta_Y"] \arrow[d, "\eta_X\times Y"'] \arrow[dr, phantom, "\lrcorner", very near start] & X \times \Sigma^2 Y \arrow[d, "\eta_X\times\Sigma^2 Y"'] \\
\Sigma^2 X \times Y \arrow[r, "\Sigma^2 X\times\eta_Y"] & \Sigma^2 X \times \Sigma^2 Y
\end{tikzcd}
\qquad
\begin{tikzcd}
A \otimes B \arrow[dr, phantom, "\lrcorner", very near start] & A^{\Sigma^B} \arrow[l] \arrow[d, dotted, shift right, "(\eta_A)^{(\Sigma^B)}"'] \\
B^{\Sigma^A} \arrow[u] \arrow[r, dotted, shift left, "(\eta_B)^{(\Sigma^A)}"] & \Sigma^{\Sigma^A\times\Sigma^B} \arrow[l, shift left, "\beta^{(\Sigma^A)}"] \arrow[u, shift right, "\alpha^{(\Sigma^B)}"']
\end{tikzcd}$$

As often happens when we consider algebras, the homomorphisms on the bottom and right are split by functions, the two idempotents in Remark 5.10 being $E_1^{\Sigma^B}$ and $E_2^{\Sigma^A}$, where $E_1 = \alpha\,;\eta_A$ and $E_2 = \beta\,;\eta_B$. When we construct the pushout, we shall find that the other two maps are also split, with the result that the pushout of functions, like the coequalisers that we have used, is ***absolute***. In particular, $\Sigma^2(A \otimes B)$ is also a pushout, enabling us to define the algebra structure $\Sigma^2(A \otimes B) \to (A \otimes B)$.

(The reason for using the symbol $\otimes$ for the coproduct of algebras is simply that the corresponding construction in locale theory is a tensor product of the underlying join semilattices.)

That is the plan for the later parts of the construction, but it also provides the idea for the central technical result. The composite $B^{\Sigma^A} \to \Sigma^{\Sigma^A\times\Sigma^B} \to A^{\Sigma^B}$ takes G to $\lambda\psi.\,\alpha\big(\lambda\phi.\,\psi(G\phi)\big)$. We now make use of the corresponding external transformation. For $A = \Sigma^X$ and $B = \Sigma^Y$, this is just the double exponential transposition $\mathcal{C}(\Sigma^{\Sigma^X}, \Sigma^Y) \cong \mathcal{C}(\Sigma^{\Sigma^Y}, \Sigma^X)$.

Lemma 7.3 $G \mapsto \Sigma^G\,;\alpha$ and $F \mapsto \Sigma^F\,;\beta$ restrict to a bijection $\mathcal{A}(\Sigma^A, B) \cong \mathbf{Alg}(\Sigma^B, A)$ that's natural with respect to homomorphisms.

$$\begin{tikzcd}
\mathcal{C}(\Sigma^B, A) \arrow[drr] & & \mathcal{C}(\Sigma^A, B) \arrow[dll] \\
\mathbf{Alg}(\Sigma^B, A) \arrow[u] \arrow[rr, shift left] \arrow[rr, phantom, "\cong"] & & \mathbf{Alg}(\Sigma^A, B) \arrow[u] \arrow[ll, shift left]
\end{tikzcd}$$

Proof. Whatever G is, $F = \Sigma^G \,;\, \alpha$ is a homomorphism, being a composite of two homomorphisms. If G is itself a homomorphism then $\Sigma^F \,;\, \beta = \Sigma^\alpha \,;\, \Sigma^{\Sigma^G} \,;\, \beta = \Sigma^{\eta_B} \,;\, G = G$. Given homomorphisms $u : A \to A'$ and $v : B \to B'$, the correspondence takes $\Sigma^v \,;\, F \,;\, u$ to $\Sigma^u \,;\, G \,;\, v$. ■

Remark 7.4 We can define an *internal* notion of "object of homomorphisms" (which should also have its own notation) by interpreting the external Eilenberg–Moore equation as the equaliser

$$\begin{tikzcd} \mathbf{Alg}\big((C,\gamma),(B,\beta)\big) \arrow[r, tail] & B^C \arrow[r, shift left, "\gamma\,;\,(-)"] \arrow[r, shift right, "\Sigma^2(-)\,;\,\beta"'] & B^{\Sigma^2 C} \end{tikzcd}$$

or by a generalisation of the equaliser in Proposition 3.2 with B in place of Σ.

The argument in the Lemma internalises, to show that $\mathbf{Alg}(\Sigma^A, B)$ is a *split* equaliser (retract) of B^{Σ^A}. Moreover, it is isomorphic in $\mathcal{C}$ to $\mathbf{Alg}(\Sigma^B, A)$, and we write $A \otimes B$ for either of them. The same argument shows that $A \otimes B$ is an absolute pushout in $\mathcal{C}$, whence it carries a unique algebra structure that makes it a pushout of algebras. ■

From the external result we can deduce Lemma 5.7, that $\mathcal{C} \to \overline{\mathcal{C}}$ preserves products of injectives (actually, just carriers of algebras, but that's no handicap).

Lemma 7.5 If $C, D \in \mathsf{ob}\mathcal{C}$ are carriers of algebras then

$$\begin{tikzcd} \Sigma^C \arrow[r, "\Sigma^{\pi_0}"] & \Sigma^{C\times D} & \Sigma^D \arrow[l, "\Sigma^{\pi_1}"'] \end{tikzcd}$$

is a coproduct diagram of algebras.

Proof. For any other algebra Θ (playing the role of Σ^Γ in Section 5),

$$\mathbf{Alg}(\Sigma^C, \Theta) \times \mathbf{Alg}(\Sigma^D, \Theta) \;\cong\; \mathbf{Alg}(\Sigma^\Theta, C) \times \mathbf{Alg}(\Sigma^\Theta, D) \;\cong\; \mathbf{Alg}(\Sigma^\Theta, C \times D) \;\cong\; \mathbf{Alg}(\Sigma^{C\times D}, \Theta)$$

using Lemma 7.3 three times. Lemma 5.5 gave the algebra structure on $C \times D$. ■

Corollary 7.6 The algebra $\Sigma^{\Sigma^A \times \Sigma^B}$ has the universal property that

$$\mathcal{C}(A, \Theta) \times \mathcal{C}(B, \Theta) \;\cong\; \mathbf{Alg}(\Sigma^{\Sigma^A \times \Sigma^B},\, \Theta).$$

Proof. Put $C = \Sigma^A$ and $D = \Sigma^B$ and recall that $\mathcal{C}(A, \Theta) \cong \mathbf{Alg}(\Sigma^{\Sigma^A}, \Theta)$. ■

Next we prove that the power algebra does the job for which it was introduced in Remark 7.1. Here A and D play the roles of $\Sigma^{\{Y|E\}}$ and W in Lemma 5.8

Lemma 7.7 For algebras A and D, the diagram $A \longrightarrow A^D \longleftarrow \Sigma^D$ is a coproduct of algebras.

$$\begin{tikzcd} A \arrow[r, dotted] & A^D & \\ \Sigma^2 A \arrow[u, "\alpha"] \arrow[r, "\Sigma^{\pi_0}"] & \Sigma^{(\Sigma^A \times D)} \arrow[u, "\alpha^D"] & \Sigma^D \arrow[l, "\Sigma^{\pi_1}"'] \arrow[ul] \arrow[dl, "\Sigma^{\pi_1}"] \\ \Sigma^4 A \arrow[u, shift left, "\Sigma^{\Sigma^\alpha}"] \arrow[u, shift right, "\Sigma^{\eta_{\Sigma A}}"'] \arrow[r, "\Sigma^{\pi_0}"] & \Sigma^{(\Sigma^3 A \times D)} \arrow[u, shift left, "\Sigma^{\Sigma^\alpha \times D}"] \arrow[u, shift right, "\Sigma^{\eta_{\Sigma A} \times D}"'] & \end{tikzcd}$$

Proof. The left-hand column is a U-split coequaliser of algebras, *i.e.* an absolute coequaliser in $\mathcal{C}$, which the functor $(-)^D$ therefore preserves. So the middle column is a coequaliser of (power) algebras, in which the middle and bottom objects are coproducts by Lemma 7.5. But the universal properties of coproducts and coequalisers commute, *cf.* products and equalisers in Lemma 5.8. ■

Corollary 7.8 The algebra A^{Σ^B} has the universal property that

$$\mathbf{Alg}(A,\Theta)\times\mathcal{C}(B,\Theta)\;\cong\;\mathbf{Alg}(A^{\Sigma^B},\,\Theta).$$ ■

Theorem 7.9 $A\otimes B$ carries the structure of the coproduct of the algebras A and B.

Proof. By Remark 7.2, $A\otimes B$ is an absolute pushout of the diagram in $\mathcal{C}$, so there is a unique algebra structure on it that makes this diagram a pushout in $\mathbf{Alg}$.

Hence we have isomorphic pullbacks in $\mathbf{Set}$,

$$\begin{tikzcd}
\mathbf{Alg}(A,\Theta)\times\mathbf{Alg}(B,\Theta) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbf{Alg}(A,\Theta)\times\mathcal{C}(B,\Theta) \arrow[d] & \mathbf{Alg}(A\otimes B,\Theta) \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbf{Alg}(A^{\Sigma^B},\Theta) \arrow[d] \\
\mathcal{C}(A,\Theta)\times\mathbf{Alg}(B,\Theta) \arrow[r] & \mathcal{C}(A,\Theta)\times\mathcal{C}(B,\Theta) & \mathbf{Alg}(B^{\Sigma^A},\Theta) \arrow[r] & \mathbf{Alg}(\Sigma^{\Sigma^A\times\Sigma^B},\Theta)
\end{tikzcd}$$

or, in plain English, if the functions $A\to\Theta$ and $B\to\Theta$ are actually both homomorphisms then they correspond to a homomorphism from $A\otimes B$. Therefore this has the universal property of the coproduct. ■

Example 7.10 Recall that any idempotent defines a partition and chooses an element from each equivalence class. In this case the two partitions are the same but the choices of elements are different. For example in the simplest case, $A=B=\mathbf{1}$, the two representations are embedded as the singletons $\{\ulcorner\pi_0\urcorner\}$ and $\{\ulcorner\pi_1\urcorner\}$ in $\Sigma^{\Sigma^{\mathbf{2}}}$.

$$\begin{tikzcd}
\mathbf{0}\times\mathbf{0} \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbf{0}\times\Sigma \arrow[d] \\
\Sigma\times\mathbf{0} \arrow[r] & \Sigma\times\Sigma
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{1} & \mathbf{1} \arrow[l] \arrow[d, shift left, "\ulcorner\pi_0\urcorner"] & F \arrow[d, mapsto] \\
\mathbf{1} \arrow[u] \arrow[r, shift right, "\ulcorner\pi_1\urcorner"'] & \Sigma^{\Sigma\times\Sigma} \arrow[l, shift right] \arrow[u, shift left] & \lambda\phi\psi.\,\phi(F\psi)
\end{tikzcd}$$

Proposition 7.11 Powers of $\overline{\Sigma}$ (as we write it) in $\overline{\mathcal{C}}\equiv\mathbf{Alg}^{\mathsf{op}}$ are given by

$$\overline{\Sigma}^{(A,\alpha)}=(\Sigma^A,\Sigma^{\eta_A})\quad\text{and}\quad\bar{\eta}_{(A,\alpha)}=\alpha,\qquad\text{where }\overline{\Sigma}=(\Sigma^\Sigma,\Sigma^{\eta_1}).$$

Proof. Since $A\otimes B$ was defined to have $\mathbf{Alg}(\Sigma^A,B)$ as its set of global elements,

$$\overline{\mathcal{C}}(\Gamma\times X,\Sigma)\;=\;\mathcal{A}(\Sigma^\Sigma,A\otimes B)\;=\;\mathcal{C}(\mathbf{1},A\otimes B)\;=\;\mathcal{A}(\Sigma^A,B)\;=\;\overline{\mathcal{C}}(\Gamma,\Sigma^X)$$

where $A=\Sigma^\Gamma$ and $B=\Sigma^X$. ■

Remark 7.12 For the other structure in terms of algebras,
(a) the terminal object is (Σ,Σ^{η_1});
(b) the natural numbers object is $\Sigma^{\mathbb{N}}$;
(c) $\overline{\overline{\mathcal{C}}}\cong\overline{\mathcal{C}}$ because $\mathcal{A}_{\overline{\mathcal{C}}}\cong\mathcal{A}_{\mathcal{C}}$.
The last part says that $\overline{\mathcal{C}}=\mathcal{A}^{\mathsf{op}}$ is monadic, and therefore admits subspaces as described in Sections 3 and 6, since we assumed that idempotents split in $\mathcal{C}$. ■

8 Comprehension

Now we develop a λ-calculus for $\overline{\mathcal{C}}$, just as Section A 8 did for $\mathsf{S}\mathcal{C}$.

The new calculus is based on a type-theoretic presentation [Tay99, Section 2.2] of Zermelo's original set theory, with Σ^X playing the role of the powerset ($\mathcal{P}X$) and the object $\{X \mid E\}$ from Section 4 that of comprehension. Whereas comprehension is often presented in a way that allows each term to belong to many types, the new operators admit, i and I in our calculus, whilst perhaps being a little bureaucratic, ensure that each term has a unique type.

Writing $\mathcal{D}$ for the category obtained from the new calculus as explained in [Tay99, Sections 4.3 & 4.7] and more briefly in Remark A 2.7, we shall show in the following sections that $\overline{\mathcal{C}} \simeq \mathcal{D}$. This will be done by means of a weak normalisation theorem, thereby showing that we have stated enough equations. It is a categorical *equivalence*, rather than the isomorphism that we had for $\mathsf{S}\mathcal{C}$, because we're now manipulating types.

Remark 8.1 In the calculus that we have in mind for *set theory*, the rules for the powerset $\mathcal{P}X \equiv \Omega^X$ are those of the restricted λ-calculus, where $\xi[a]$ means $a \in \{X \mid \xi\}$. Then comprehension has formation and introduction rules,

$$\frac{X\ \mathsf{type} \qquad \vdash \xi : \mathcal{P}X}{\{X \mid \xi\}\ \mathsf{type}} \qquad\qquad \frac{\Gamma \ \vdash\ a : X \qquad \Gamma \ \vdash\ \xi[a]}{\Gamma \ \vdash\ \mathsf{admit}\, a : \{X \mid \xi\}}$$

elimination rules,

$$x : \{X \mid \xi\} \ \vdash\ i_{X,\xi} x : X \qquad\qquad x : \{X \mid \xi\} \ \vdash\ \xi[i_{X,\xi} x]$$

and β and η-rules

$$\frac{\Gamma \ \vdash\ a : X \qquad \Gamma \ \vdash\ \xi[a]}{\Gamma \ \vdash\ a = i_{X,\xi}(\mathsf{admit}\, a) : X} \qquad x : \{X \mid \xi\} \ \vdash\ x = \mathsf{admit}(i_{X,\xi} x).$$

The intent of these rules is to specify when there is a term $\mathsf{admit}\, a$ that makes the triangle commute:

$$\begin{tikzcd} & \Gamma \arrow[dl, dotted, "\mathsf{admit}\, a"'] \arrow[dr, "a"] & \\ \{X \mid \xi\} \arrow[rr, tail, "i_{X,\xi}"'] & & X \end{tikzcd}$$

Notice that we have required $\xi : \mathcal{P}X$ to be a *closed* term (defined in the empty context), because we do not want to get involved in dependent types in this paper.

Remark 8.2 The rules that we give below for our own monadic situation are not very pretty if seen as a piece of pure λ-calculus. But they are in the spirit of the treatment of comprehension above, so long as we read the the type-formation rule as

$$\frac{X\ \mathsf{type} \qquad\qquad \text{subtype data on } X}{\{X \mid \text{subtype data}\}\ \mathsf{type}}$$

where any predicate $\xi : \mathcal{P}X$ provides subtype data in set theory, and the nucleus E (Definition 4.3) does so for our own calculus. Similarly, the introduction rule means

$$\frac{\Gamma \ \vdash\ a : X \qquad\qquad \Gamma \ \vdash\ a \text{ is admissible to the subtype}}{\Gamma \ \vdash\ \mathsf{admit}\, a : \{X \mid \text{subtype data}\}}$$

where our admissibility condition is unfortunately more complicated and less intuitive than that in set theory.

You may think of

$$\{X \mid E\} \qquad \text{as} \qquad \{x : X \mid \forall\phi{:}\Sigma^X.\ E\phi x =_\Sigma \phi x\},$$

but this can only be internalised if Σ^X is compact (so $\forall\phi : \Sigma^X$ is meaningful) and Σ is discrete (so $=_\Sigma$ is defined). These conditions (along with what is needed in Example 2.6(a) to define $I_{X,E}\theta$ in terms of $\exists$) would make $\overline{\mathcal{C}}$ into a topos (Theorem C 11.12), and thereby bring the comprehension calculus back to that for Zermelo type theory.

Remark 8.3 As we have learned throughout this paper, we need to be explicit about the subspace topology. The elimination rule for elements ($\{\}E_0$) below provides the restriction of an open subset ϕ of X to the subspace $\{X \mid E\}$, which we may re-interpret as the introduction rule for predicates,

$$\frac{\Gamma \ \vdash \ \phi : \Sigma^X}{\Gamma \ \vdash \ \Sigma^i\phi \equiv \lambda x.\ \phi(i_{X,E}x) : \Sigma^{\{X|E\}}}\ \Sigma^{\{\}}I$$

For the corresponding elimination rule, we need a new operator $I_{X,E}$ to expand an open subset θ of the subspace to the whole space.

$$\begin{tikzcd} \Gamma\times\{X \mid E\} \arrow[rr, tail, "\Gamma\times i_{X,E}"] \arrow[dr, "\theta"'] & & \Gamma\times X \arrow[dl, dotted, "I_{X,E}\theta"] \\ & \Sigma & \end{tikzcd}$$

For reference, we repeat Definitions A 2.1, A 8.1, A 8.2 and 4.3.

Definition 8.4 The ***restricted λ-calculus*** has just the type-formation rules

$$\mathbf{1}\ \mathsf{type} \qquad \frac{X_1\ \mathsf{type} \quad \ldots X_k\ \mathsf{type}}{\Sigma^{X_1\times\cdots\times X_k}\ \mathsf{type}}\ \Sigma^{(-)}F$$

but with the normal rules for λ-abstraction and application,

$$\frac{\Gamma, x : X \ \vdash \ \sigma : \Sigma^Y}{\Gamma \ \vdash \ \lambda x{:}X.\ \sigma : \Sigma^{X\times Y}}\ \Sigma^{(-)}I \qquad \frac{\Gamma \ \vdash \ \phi : \Sigma^{X\times Y} \quad \Gamma \ \vdash \ a : X}{\Gamma \ \vdash \ \phi[a] : \Sigma^Y}\ \Sigma^{(-)}E$$

together with the usual α, β and η rules.

Definition 8.5 $\Gamma \vdash P : \Sigma^{\Sigma^X}$ is ***prime*** if $\quad \Gamma,\ \mathcal{F} : \Sigma^3X \ \vdash \ \mathcal{F}P \ = \ P\big(\lambda x.\ \mathcal{F}(\lambda\phi.\ \phi x)\big)$.

Definition 8.6 The ***sober λ-calculus*** has the additional rules

$$\frac{\Gamma \ \vdash \ P : \Sigma^{\Sigma^X} \qquad P \text{ is prime}}{\Gamma \ \vdash \ \mathsf{focus}\,P : X} \qquad \mathsf{focus}\,I$$

$$\frac{\Gamma \ \vdash \ P : \Sigma^{\Sigma^X} \qquad P \text{ is prime}}{\Gamma,\ \phi : \Sigma^X \ \vdash \ \phi(\mathsf{focus}\,P) = P\phi : \Sigma} \qquad \mathsf{focus}\,\beta$$

$$\frac{\Gamma \ \vdash \ a, b : X \qquad \Gamma,\ \phi : \Sigma^X \ \vdash \ \phi a = \phi b}{\Gamma \ \vdash \ a = b} \qquad \mathrm{T}_0$$

Definition 8.7 $\phi : \Sigma^Y \vdash E\phi : \Sigma^Y$ is called a ***nucleus*** if

$$\mathcal{F} : \Sigma^3 Y \;\;\vdash\;\; E\big(\lambda y.\, \mathcal{F}(\lambda\phi.\, E\phi y)\big) \;=\; E\big(\lambda y.\, \mathcal{F}(\lambda\phi.\, \phi y)\big).$$

Definition 8.8 The $\{\}$-rules of the ***monadic*** $\boldsymbol{\lambda}$***-calculus*** define the subspace itself.

$$\frac{X\ \mathsf{type} \qquad x : X,\ \phi : \Sigma^X \ \vdash\ E\phi x : \Sigma \qquad E \text{ is a nucleus}}{\{X \mid E\}\ \mathsf{type}} \qquad \{\}F$$

$$\frac{\Gamma \ \vdash\ a : X \qquad \Gamma,\ \phi : \Sigma^X \ \vdash\ \phi a = E\phi a}{\Gamma \ \vdash\ \mathsf{admit}_{X,E}\, a : \{X \mid E\}} \qquad \{\}I$$

$$x : \{X \mid E\} \ \vdash\ i_{X,E} x : X \qquad \{\}E_0$$

$$x : \{X \mid E\},\ \phi : \Sigma^X \ \vdash\ \phi(i_{X,E}x) = E\phi(i_{X,E}x) \qquad \{\}E_1$$

$$\frac{\Gamma \ \vdash\ a : X \qquad \Gamma,\ \phi : \Sigma^X \ \vdash\ \phi a = E\phi a}{\Gamma \ \vdash\ a = i_{X,E}(\mathsf{admit}_{X,E}\, a) : X} \qquad \{\}\beta$$

$$x : \{X \mid E\} \ \vdash\ x = \mathsf{admit}_{X,E}(i_{X,E}x) \qquad \{\}\eta$$

Definition 8.9 The $\Sigma^{\{\}}$-rules say that it has the subspace topology, where $I_{X,E}$ expands open subsets of the subspace to the whole space.

$$\theta : \Sigma^{\{X|E\}} \ \vdash\ I_{X,E}\theta : \Sigma^X \qquad \Sigma^{\{\}}E$$

The β-rule says that the composite $\Sigma^X \twoheadrightarrow \Sigma^{\{X|E\}} \rightarrowtail \Sigma^X$ is E:

$$\phi : \Sigma^X \ \vdash\ I_{X,E}\big(\lambda x{:}\{X \mid E\}.\ \phi(i_{X,E}x)\big) = E\phi \qquad \Sigma^{\{\}}\beta$$

Notice that this is the only rule that introduces E into the λ-expressions. The η-rule, which we first saw in Remark 2.8, says that the other composite $\Sigma^{\{X|E\}} \rightarrowtail \Sigma^X \twoheadrightarrow \Sigma^{\{X|E\}}$ is the identity:

$$\theta : \Sigma^{\{X|E\}},\ x : \{X \mid E\} \ \vdash\ \theta x = I_{X,E}\theta(i_{X,E}x) \qquad \Sigma^{\{\}}\eta.$$

In any type theory where terms are embedded in the definitions of types, we must check that, when two terms are equal according to the rules of the calculus, they give rise to interchangeable types. This result has the flavour of the normalisation proof that follows (Proposition 9.11), whilst being a lot simpler, so you may find it helpful to fill in the details of the verifications as a preparatory exercise.

Lemma 8.10 If $E_1 = E_2$ then there is an isomorphism $\{X \mid E_1\} \cong \{X \mid E_2\}$ that commutes with the structure (admit, i and I).

PROOF. The isomorphism for values is $x_1 \mapsto \mathsf{admit}_2(i_1 x_1)$ and *vice versa*. This is well formed, *i.e.* the side condition for $\{\}I$ is satisfied and justifies the use of admit_2, because of the equation $E_1 = E_2$. The isomorphism for properties is $\theta_1 \mapsto \Sigma^{i_2}(I_1\theta_1)$. The equations for the isomorphisms and commutativity with the structure follow easily from the β- and η-rules for the two sides. ■

Before embarking on the main normalisation theorem, we investigate some of the interactions between the new admit operation and the underlying sober and recursive structures. The first result is the symbolic analogue of Remark 4.11, showing that admit does the same for a general Σ-split inclusion as focus does for η.

Lemma 8.11 The standard resolution defines an isomorphism

$$X \underset{\mathsf{focus}\cdot i}{\overset{\mathsf{admit}\cdot\eta_X}{\underset{\longleftarrow}{\overset{\longrightarrow}{\cong}}}} \{\Sigma^2 X \mid E\} \qquad \text{where } E = \eta_{\Sigma^X}\cdot\Sigma^{\eta_X}.$$

Proof. The term $x : X \vdash \mathsf{admit}(\lambda\phi.\,\phi x)$ is well formed because of the unit law. The right hand side of Definition 8.5 is $E\mathcal{F}P$, so the subspace $\{\Sigma^2 X \mid E\}$ exactly captures the primes and

$$P : \{\Sigma^2 X \mid E\} \;\vdash\; \mathsf{focus}(iP) : X$$

is well formed. Then $\mathsf{focus}\big(i(\mathsf{admit}(\lambda\phi.\,\phi x))\big) \;=\; \mathsf{focus}(\lambda\phi.\,\phi x) \;=\; x$ by $\{\}\beta$ and $\mathsf{focus}\,\eta$. Conversely, $\mathsf{admit}\big(\eta_X(\mathsf{focus}(iP))\big) \;=\; \mathsf{admit}(iP) \;=\; P$ by $\mathsf{focus}\,\beta$ and $\{\}\eta$. ∎

The interaction between admit and substitution, *i.e.* cut elimination, is as expected, *cf.* Lemma A 8.4.

Lemma 8.12 Let $\Gamma \vdash \mathsf{admit}_{X,E}\, a : \{X \mid E\}$ be a well formed term and $u : \Delta \to \Gamma$ a substitution [Tay99, Section 4.3]. Then $\Delta \vdash \mathsf{admit}_{X,E}\, u^* a : \{X \mid E\}$ is also well formed, and

$$\Delta \;\vdash\; u^*(\mathsf{admit}_{X,E}\, a) \;=\; \mathsf{admit}_{X,E}(u^* a).$$

Proof. In the context $[\Delta,\ \phi : \Sigma^X]$, since E and ϕ do not depend on Γ,

$$\phi(u^* a) \;\equiv\; u^*(\phi a) \;=\; u^*(E\phi a) \;\equiv\; E\phi(u^* a).$$

This is consistent with the $\{\}\beta$- and η-rules when we put $u^*(i_{X,E} a) = i_{X,E}(u^* a)$, and with the $\Sigma^{\{\}}$-rules since $u^*(I_{X,E}\theta) = I_{X,E}(u^*\theta)$. ∎

Much of the business of Section A 8, which introduced the focus operation to handle sobriety, was to *eliminate* it. In particular, Proposition A 8.10 showed that focus is redundant for logical terms (*i.e.* of type Σ^U), whilst Sections A 9–10 replaced it with descriptions for numerical terms (*i.e.* of type $\mathbb{N}$). The following result adds a new case to that reduction, for terms of type $\{Y \mid E\}$, so focus can still be pushed to the outside of any term of the comprehension calculus, or eliminated altogether.

Lemma 8.13 If $\Gamma \vdash P : \Sigma^2\{Y \mid E\}$ is prime then

$$\mathsf{focus}_{\{Y\mid E\}}\, P \;=\; \mathsf{admit}_{Y,E}\,\big(\mathsf{focus}_Y(\Sigma^2 i_{Y,E} P)\big).$$

So $\{Y \mid E\}$ is sober, *cf.* Lemma 6.9.

Proof. Since $\Gamma \vdash \Sigma^2 iP : \Sigma^2 Y$ is also prime (Lemma A 4.6), we may form

$$\Gamma \;\vdash\; a \;\equiv\; \mathsf{focus}_Y(\Sigma^2 iP) : Y,$$

which satisfies

$$\begin{array}{rcll}
\phi a & = & \phi\big(\mathsf{focus}_Y(\Sigma^2 iP)\big) & \\
& = & \Sigma^2 iP\phi & \mathsf{focus}\,\beta \\
& = & P(\Sigma^i\phi) \;=\; P\big(\lambda y.\,\phi(iy)\big) & \text{definition of } \Sigma^2 i \\
& = & P\big(\lambda y.\,E\phi(iy)\big) & \{\}E_1 \\
& = & E\phi a & \text{in the same way,}
\end{array}$$

so we may then form $\Gamma \vdash \mathsf{admit}_{Y,E}\left(\mathsf{focus}_Y(\Sigma^2 iP)\right)$. Applying $\theta : \Sigma^{\{X|E\}}$,

$$\begin{array}{lll} \theta\big(\mathsf{admit}\left(\mathsf{focus}_Y(\Sigma^2 iP)\right)\big) & & \\ \quad = & I\theta\big(\mathsf{focus}_Y(\Sigma^2 iP)\big) & \{\}\beta,\ \Sigma^{\{\}}\eta \\ \quad = & \Sigma^2 iP(I\theta) & \mathsf{focus}\,\beta \\ \quad = & P\big(\Sigma^i(I\theta)\big) \ = \ P\theta & \Sigma^{\{\}}\eta \\ \quad = & \theta(\mathsf{focus}_{\{Y|E\}}\,P) & \mathsf{focus}\,\beta, \end{array}$$

whence the result follows by T_0. ∎

Finally, admit can also be moved to the outside of the term, although that is in fact the normalisation theorem to which the next two sections are devoted. Here we just consider the interaction with recursion, which is the symbolic analogue of Proposition 6.14.

Lemma 8.14 Let $X = \{Y \mid E\}$ with terms

$$\Gamma \ \vdash \ n : \mathbb{N}, \qquad \Gamma \ \vdash \ z : Y \qquad \text{and} \qquad \Gamma,\, m : \mathbb{N},\, y : Y \ \vdash \ s(m,y) : Y$$

such that $\Gamma \vdash \mathsf{admit}\, z : X$ and $\Gamma,\ m : \mathbb{N},\ x : X \vdash \mathsf{admit}\, s(m, ix) : X$ are well formed. Then

$$\Gamma \ \vdash \ \mathsf{rec}\big(n,\ \mathsf{admit}\, z,\ \lambda mx.\ \mathsf{admit}\, s(m,ix)\big) \ = \ \mathsf{admit}\,\mathsf{rec}\big(n,\ z,\ \lambda my.\ s(m,y)\big) : X,$$

omitting the subscripts on $i_{Y,E}$ and $\mathsf{admit}_{Y,E}$.

Proof. Using the $\{\}\beta$- and η-rules, the claim is equivalent to

$$\Gamma \ \vdash \ i\,\mathsf{rec}\big(n,\ \mathsf{admit}\, z,\ \lambda mx.\ \mathsf{admit}\, s(m,ix)\big) \ = \ \mathsf{rec}\big(n,\ z,\ \lambda my.\ s(m,y)\big) : Y$$

and is valid for $n = 0$. In order to use the $\mathsf{rec}\,\eta$-rule to prove the claim, we have to check the same equation as for the induction step in an ordinary proof by induction, namely

$$\begin{array}{lll} i\,\mathsf{rec}\big(n+1,\ \mathsf{admit}\, z,\ \lambda mx.\ \mathsf{admit}\, s(m,ix)\big) & & \text{recursion step} \\ \quad = & i\,\mathsf{admit}\, s\big(n, i\,\mathsf{rec}\big(n,\ \mathsf{admit}\, z,\ \lambda mx.\ \mathsf{admit}\, s(m,ix)\big)\big) & \\ \quad = & i\,\mathsf{admit}\, s\big(n, \mathsf{rec}\big(n,\ z,\ \lambda my.\ s(m,y)\big)\big) & \text{induction hypothesis} \\ \quad = & s\big(n, \mathsf{rec}\big(n,\ z,\ \lambda my.\ s(m,y)\big)\big) & \{\}\beta \\ \quad = & \mathsf{rec}\big(n+1,\ z,\ \lambda my.\ s(m,y)\big). & \text{recursion step.} \end{array}$$ ∎

9 Normalisation for types

This section shows how each new type (generated arbitrarily from comprehension and powers of products) may be expressed as a subtype (by a single comprehension) of a type in the original calculus. This is like expressing any set in Zermelo set theory as a subset of an iterated powerset, as in the von Neumann hierarchy. It will relate back to the categorical construction in Section 4, where every new object of $\overline{\mathcal{C}}$ is a formal equaliser in $\mathcal{C}$.

The structure maps in this isomorphism are, of course, *terms to be defined*, but they *behave* like the constructors i, I and admit of the comprehension calculus. That is, the lemmas that we have to prove towards this isomorphism are like the β- and η-rules of the calculus, and will therefore be presented and named as such. This is not a serious overloading of the i and I notations, as the old use has two subscripts.

Remark 9.1 Given any λ-calculus (such as ours) with all of the usual structural rules, the corresponding category has products simply because they are concatenations of contexts. The *problem* with products that was a central concern in the two categorical constructions does not, unfortunately, go away so easily: we have merely pushed it into the exponents.

A power type is $\Sigma^{(-)}$ applied to (the product of) a *list* of types, and λ-abstraction and application involve the list operations of *push* and *pop* (or *cons* and (*head,tail*)) on types. The normalisation process for types must therefore also deal with lists. This could be formalised by introducing meta-variables to denote either lists of types, or powers of lists. However, as this complication does not have any impact on the real issues in the argument, we leave the interested reader to carry this through for themselves, and simply consider the case of lists of length two. We take $\Sigma^{Y\times Z}$ to mean $(\Sigma^Z)^Y$, a typical term of which is $\lambda y.\,\phi$, where $y : Y$ and $\phi : \Sigma^Z$. In this sense, Σ contains the empty list.

The main technical question with products then manifests itself in this situation as the choice between $I_Z^{\overline{Y}} \cdot I_Y^Z$ and the other term that could serve as $i_{\Sigma^{Y\times Z}}$.

Extended notation

Notation 9.2 By structural recursion on the type X, we define a type $\overline{X}$, terms

$$X \underset{\mathsf{admit}_X}{\overset{i_X}{\rightleftarrows}} \overline{X} \qquad\qquad \Sigma^X \underset{\Sigma^{i_X}}{\overset{I_X}{\rightleftarrows}} \Sigma^{\overline{X}}$$

and the idempotent E_X on $\Sigma^{\overline{X}}$. The base cases are $\mathbf{1}$, Σ and $\mathbb{N}$, for which these maps are identities.

X	$\Sigma^{Y\times Z}$	$\{Y \mid E\}$
$\overline{X}$	$\Sigma^{\overline{Y}\times\overline{Z}}$	$\overline{Y}$
i_X	$\Sigma^{Y\times Z} \xrightarrow{I_Y^Z} \Sigma^{\overline{Y}\times Z} \xrightarrow{I_Z^{\overline{Y}}} \Sigma^{\overline{Y}\times\overline{Z}}$	$\{Y \mid E\} \xrightarrow{i_{Y,E}} Y \xrightarrow{i_Y} \overline{Y}$
admit_X	$\Sigma^{\overline{Y}\times\overline{Z}} \xrightarrow{\Sigma^{i_Y\times i_Z}} \Sigma^{Y\times Z}$	$\overline{Y} \xrightarrow{\mathsf{admit}_Y} Y \xrightarrow{\mathsf{admit}_{Y,E}} \{Y \mid E\}$
I_X	$\Sigma^2(Y\times Z) \xrightarrow{\Sigma^2(i_Y\times i_Z)} \Sigma^2(\overline{Y}\times\overline{Z})$	$\Sigma^{\{Y\mid E\}} \xrightarrow{I_{Y,E}} \Sigma^Y \xrightarrow{I_Y} \Sigma^{\overline{Y}}$
E_X	$\Sigma^2(\overline{Y}\times\overline{Z}) \xrightarrow{\Sigma(E_Z^{\overline{Y}}\cdot E_Y^{\overline{Z}})} \Sigma^2(\overline{Y}\times\overline{Z})$	$\Sigma^{\overline{Y}} \xrightarrow{\Sigma^{i_Y}} \Sigma^Y \xrightarrow{E} \Sigma^Y \xrightarrow{I_Y} \Sigma^{\overline{Y}}$

So $\overline{X}$ erases the comprehensions from X, and i_X embeds X as a subspace of $\overline{X}$. This has the subspace topology because of I_X.

Lemma 10.10 shows that $E_{\{Y\mid E\}} = \overline{E}$, where the translation $\overline{E}$ for terms is defined in the next section.

The normalised type

Lemma 9.3 The $\Sigma^{\{\}}\eta$ rule, $\Sigma^{i_X} \cdot I_X = \mathsf{id}_{\Sigma^X}$ or $\phi : \Sigma^X, x : X \vdash (I_X\phi)(i_X x) = \phi x$, is valid.

PROOF. By structural induction on the type X. If $X = \Sigma^{Y\times Z}$ then

$$\begin{aligned}
\Sigma^{i_X} \cdot I_X &= \Sigma(I_Z^{\overline{Y}} \cdot I_Y^Z) \cdot \Sigma^2(i_Y \times i_Z) \\
&= \Sigma\big(\Sigma^{i_Y\times Z} \cdot (\Sigma^{i_Z} \cdot I_Z)^{\overline{Y}} \cdot I_Y^Z\big) && \Sigma^{i_Y\times i_Z} = \Sigma^{i_Y\times Z} \cdot (\Sigma^{i_Z})^{\overline{Y}} \\
&= \Sigma\big((\Sigma^{i_Y} \cdot I_Y)^Z\big) && \text{induction hypothesis for } Z \\
&= \mathsf{id} && \text{induction hypothesis for } Y.
\end{aligned}$$

If $X = \{Y \mid E\}$ then

$$\begin{aligned}
\Sigma^{i_X} \cdot I_X &= \Sigma^{i_{Y,E}} \cdot \Sigma^{i_Y} \cdot I_Y \cdot I_{Y,E} \\
&= \Sigma^{i_{Y,E}} \cdot I_{Y,E} && \text{induction hypothesis for } Y \\
&= \mathsf{id} && \Sigma^{\{\}}\eta \text{ for } \{Y \mid E\}. \blacksquare
\end{aligned}$$

Lemma 9.4 The $\Sigma^{\{\}}\beta$ rule, $I_X \cdot \Sigma^{i_X} = E_X$, is valid.

PROOF. By structural induction on the type X. If $X = \Sigma^{Y\times Z}$ then

$$\begin{aligned}
I_X \cdot \Sigma^{i_X} &= \Sigma^2\big(i_Y \times i_Z\big) \cdot \Sigma(I_Z^{\overline{Y}} \cdot I_Y^Z\big) \\
&= \Sigma\big(I_Z^{\overline{Y}} \cdot (I_Y \cdot \Sigma^{i_Y})^Z \cdot \Sigma^{\overline{Y}\times i_Z}\big) \\
&= \Sigma(I_Z^{\overline{Y}} \cdot E_Y^Z \cdot \Sigma^{\overline{Y}\times i_Z}) && \text{induction hypothesis for } Y \\
&= \Sigma\big((I_Z \cdot \Sigma^{i_Z})^{\overline{Y}} \cdot E_Y^{\overline{Z}}\big) && E_Y^{(-)} \text{ natural w.r.t. } i_Z \\
&= \Sigma(E_Z^{\overline{Y}} \cdot E_Y^{\overline{Z}}) && \text{induction hypothesis for } Z.
\end{aligned}$$

If $X = \{Y \mid E\}$ then

$$\begin{aligned}
I_X \cdot \Sigma^{i_X} &= I_Y \cdot I_{Y,E} \cdot \Sigma^{i_{Y,E}} \cdot \Sigma^{i_Y} \\
&= I_Y \cdot E \cdot \Sigma^{i_Y} && \Sigma^{\{\}}\beta \text{ for } \{Y \mid E\}. \blacksquare
\end{aligned}$$

Proposition 9.5 $\{\overline{X} \mid E_X\}$ is a well formed type, with structure i_X and I_X.

PROOF. E_X on $\overline{X}$ is a nucleus (Definition 8.7), by the same argument as we used in Corollary 4.13. ■

The other rules for the extended notation

Lemma 9.6 The $\{\}E_0$ rule is well formed and the $\{\}E_1$ rule is valid.

PROOF. There is no issue of well-formedness, as i_X at any type is built using $i_{Y,E}$ and $I_{Y,E}$ but not $\mathsf{admit}_{Y,E}$ from the comprehension calculus. Now let $x : X$ and $\phi : \Sigma^{\overline{X}}$. From Lemmas 9.3 and 9.4 we have

$$\Sigma^{i_X} \cdot E_X \;=\; \Sigma^{i_X} \cdot I_X \cdot \Sigma^{i_X} \;=\; \Sigma^{i_X},$$

so $\phi(i_X x) \;=\; \Sigma^{i_X}\phi x \;=\; (\Sigma^{i_X} \cdot E_X)\phi x \;=\; E_X\phi(i_X x)$. ■

Lemma 9.7 The $\{\}I$ rule is well formed, and the $\{\}\beta$ rule is valid, at type $X = \Sigma^{Y\times Z}$.

$$\frac{\Gamma \ \vdash \ \phi : \overline{X} \equiv \Sigma^{\overline{Y}\times\overline{Z}} \qquad \Gamma,\ F : \Sigma^{\overline{X}} \equiv \Sigma^2(\overline{Y}\ \times\ \overline{Z}) \ \vdash \ F\phi = E_X F\phi}{\Gamma \ \vdash \ i_X(\mathsf{admit}_X\,\phi) \ = \ \phi : X \ \equiv \ \Sigma^{Y\times Z}}$$

PROOF. In the introduction rule, $\mathsf{admit}_X\,\phi = \Sigma^{i_Y\times i_Z}\phi$ with no issue of well-formedness, as this expression does not use admit from the comprehension calculus. Also,

$$\begin{array}{rcll} F\phi & = & E_X F\phi & \text{given side condition} \\ & = & I_X(\Sigma^{i_X}F)\phi & \text{Lemma 9.4} \\ & = & \Sigma^{\mathsf{admit}_X}(\Sigma^{i_X}F)\phi & \\ & = & F(i_X\,\mathsf{admit}_X\,\phi). & \end{array}$$

So $\phi = i_X\,\mathsf{admit}_X\,\phi$ by T_0. ■

Lemma 9.8 If $X = \{Y \mid E\}$ then $\widehat{E_X} \subset_{\overline{Y}} \widehat{E_Y}$ in the sense of Notation 4.4.

PROOF. $E_X = E_{\{Y|E\}} = I_Y\cdot E\cdot\Sigma^{i_Y}$, whilst $E_Y = I_Y\cdot\Sigma^{i_Y}$ by Lemma 9.4. Then

$$E_Y\cdot E_X \ = \ I_Y\cdot\Sigma^{i_Y}\cdot I_Y\cdot E\cdot\Sigma^{i_Y} \ = \ I_Y\cdot E\cdot\Sigma^{i_Y} \ = \ E_X \ = \ I_Y\cdot E\cdot\Sigma^{i_Y}\cdot I_Y\cdot\Sigma^{i_Y} \ = \ E_X\cdot E_Y$$

by Lemma 9.3 ($\Sigma^{\{\}}\eta$ for Y). ■

Lemma 9.9 The $\{\}I$ rule is well formed, and the $\{\}\beta$ rule is valid, at all types.

PROOF. By structural induction on the type X, with $Y = \mathbb{N}$ or Σ^Z as the base case (even when Z is itself a product or comprehension type), so consider $X = \{Y \mid E\}$.

$$\frac{\Gamma \ \vdash \ a : \overline{X} \qquad \Gamma,\ \phi : \Sigma^{\overline{X}} \ \vdash \ \phi a = E_X\phi a}{\Gamma \ \vdash \ a \ = \ i_X(\mathsf{admit}_X\,a) \ \equiv \ i_Y\big(i_{Y,E}(\mathsf{admit}_{Y,E}(\mathsf{admit}_Y\,a))\big) : X}$$

First, $\phi a \ = \ E_X\phi a \ = \ (E_Y\cdot E_X)\phi a \ = \ E_Y\phi a$ using the given side-condition twice, and since $\widehat{E_X} \subset \widehat{E_Y}$. Hence $b \equiv \mathsf{admit}_Y\,a : Y$ is well formed, and $i_Y b = a$ by the induction hypothesis, $\{\}\beta$ for Y.

For the whole expression to be well formed, we need $\psi : \Sigma^Y \vdash \psi b = E\psi b$. Put $\phi = I_Y\psi : \Sigma^{\overline{Y}}$. Then

$$\begin{array}{rcll} \psi b & = & (I_Y\psi)(i_Y b) & \Sigma^{\{\}}\eta \text{ for } Y \\ & = & \phi a & \\ & = & E_X\phi\,a & \text{hypothesis} \\ & = & (I_Y\cdot E\cdot\Sigma^{i_Y})\phi\,a & \text{definition of } E_X \\ & = & (I_Y\cdot E\cdot\Sigma^{i_Y})\phi\,(i_Y b) & \\ & = & (E\cdot\Sigma^{i_Y})\phi\,b & \Sigma^{\{\}}\eta \text{ for } Y \\ & = & (E\cdot\Sigma^{i_Y}\cdot I_Y)\psi\,b & \\ & = & E\psi\,b & \Sigma^{\{\}}\eta \text{ for } Y. \end{array}$$

Finally, $i_X(\mathsf{admit}_X\,a) \ = \ i_{Y,E}\big(i_Y(\mathsf{admit}_Y(\mathsf{admit}_{Y,E}\,a))\big) \ = \ i_{Y,E}(\mathsf{admit}_{Y,E}\,a)$ by the induction hypothesis, but this is a by $\{\}\beta$ for $\{Y \mid E\}$. ■

Lemma 9.10 The $\{\}\eta$ rule, $\quad \mathsf{admit}_X \cdot i_X = \mathsf{id}_X, \quad$ is valid.

Proof. The expression is well formed by Lemmas 9.6 and 9.9. We prove the equation by structural induction on the type X. If $X = \Sigma^{Y \times Z}$ then, as in Lemma 9.3,

$$\begin{aligned} \mathsf{admit}_X \cdot i_X &= \Sigma^{i_Y \times Z} \cdot \Sigma^{\overline{Y} \times i_Z} \cdot I_Z^{\overline{Y}} \cdot I_Y^Z && \\ &= \Sigma^{i_Y \times \overline{Z}} \cdot I_Y^Z && \text{induction hypothesis for } Z \\ &= \mathsf{id} && \text{induction hypothesis for } Y. \end{aligned}$$

If $X = \{Y \mid E\}$ then $\mathsf{admit}_X \cdot i_X = \mathsf{admit}_{Y,E} \cdot \mathsf{admit}_Y \cdot i_Y \cdot i_{Y,E}$, which is $\mathsf{admit}_{Y,E} \cdot i_{Y,E}$ by the induction hypothesis, but this is $\mathsf{id}_{\{Y \mid E\}}$ by $\{\}\eta$ for $\{Y \mid E\}$. ■

The isomorphism

Proposition 9.11

$$\begin{tikzcd} X \arrow[r, shift left=2, "x \mapsto \mathsf{admit}_{\overline{X}, E_X}(i_X x)"] \arrow[r, phantom, "\cong"] & \{\, \overline{X} \mid E_X\} \arrow[l, shift left=2, "\mathsf{admit}_X(i_{\overline{X}, E_X}\, \overline{x}\,) \mapsfrom \overline{x}"] \end{tikzcd}$$

Proof. For the two admit-expressions to be well formed, we need properties of the form $E_X \phi(ix) = \phi(ix)$, where ix is either $i_X x$ or $i_{\overline{X}, E_X}\, \overline{x}$, *cf.* Lemma 8.10.

$\mathsf{admit}_{\overline{X}, E_X}(i_X x)$ is well formed by Lemma 9.6 ($\{\}E_1$ for X) and $\{\}I$ for $\{\, \overline{X} \mid E_X\}$.

$\mathsf{admit}_X(i_{\overline{X}, E_X}\, \overline{x}\,)$ is well formed by $\{\}E_1$ for $\{\, \overline{X} \mid E_X\}$ and Lemma 9.9 ($\{\}I$ for X).

$\mathsf{admit}_X(i_{\overline{X}, E_X}(\mathsf{admit}_{\overline{X}, E_X}(i_X x))) = \mathsf{admit}_X(i_X x)$ by $\{\}\beta$ for $\{\, \overline{X} \mid E_X\}$, but this is x by Lemma 9.10 ($\{\}\eta$ for X).

$\mathsf{admit}_{\overline{X}, E_X}(i_X(\mathsf{admit}_X(i_{\overline{X}, E_X}\, \overline{x}\,))) = \mathsf{admit}_{\overline{X}, E_X}(i_{\overline{X}, E_X}\, \overline{x}\,)$ by Lemma 9.9 ($\{\}\beta$ for X), but this is $\overline{x}$ by $\{\}\eta$ for $\{\, \overline{X} \mid E_X\}$. ■

Proposition 9.12 This induces $\begin{tikzcd} \Sigma^X \arrow[r, shift left=2, "\phi \mapsto \lambda\, \overline{x}\,.\, I_X \phi(i_{\overline{X}, E_X}\, \overline{x}\,)"] \arrow[r, phantom, "\cong"] & \Sigma^{\{\, \overline{X} \mid E_X\}} \arrow[l, shift left=2, "\lambda x.\, I_{\overline{X}, E_X} \phi(i_X x) \mapsfrom \theta"] \end{tikzcd}.$

Proof. $\phi\big(\mathsf{admit}_X(i_{\overline{X}, E_X}\, \overline{x}\,)\big) = I_X \phi\big(i_X(\mathsf{admit}_X(i_{\overline{X}, E_X}\, \overline{x}\,))\big) = I_X \phi(i_{\overline{X}, E_X}\, \overline{x}\,)$ because of Lemmas 9.3 and 9.9 ($\Sigma^{\{\}}\eta$ and $\{\}\beta$ for X).

$\theta\big(\mathsf{admit}_{\overline{X}, E_X}(i_X x)\big) = I_{\overline{X}, E_X} \theta\big(i_{\overline{X}, E_X}(\mathsf{admit}_{\overline{X}, E_X}(i_X x))\big) = I_{\overline{X}, E_X} \theta(i_X x)$ by $\Sigma^{\{\}}\eta$ and $\{\}\beta$ for $\{\, \overline{X} \mid E_X\}$. ■

10 Normalisation for terms

In the previous section we defined the translation $\overline{X}$ of each type X simply by *erasing* comprehensions. Although the inclusion map $i_X : X \rightarrowtail \overline{X}$ was more complicated, we shall now see that the corresponding translation $\bar{a}$ of terms is equally simple: erase the connectives i and admit, but replace (the in any case unnatural) I with $\overline{E}$.

Notation 10.1 By structural recursion on terms, we define the translation

$$\begin{array}{rcl@{\qquad\qquad}rcl}
\overline{\top} & = & \top & \overline{\bot} & = & \bot \\
\overline{0} & = & 0 & \overline{\mathsf{succ} n} & = & \mathsf{succ}\,\bar{n} \\
\overline{\phi \wedge \psi} & = & \bar{\phi} \wedge \bar{\psi} & \overline{\phi \vee \psi} & = & \bar{\phi} \vee \bar{\psi} \\
\overline{\lambda x.\, \phi} & = & \lambda\, \bar{x}\, .\ \bar{\phi} & \overline{\phi a} & = & \bar{\phi}\ \bar{a} \\
\overline{\mathsf{admit}_{X,E}\, a} & = & \bar{a} & \overline{i_{X,E} a} & = & \bar{a} \\
\overline{I_{X,E} \theta} & = & \overline{E}\ \bar{\theta} & \overline{\exists n.\, \phi[n]} & = & \exists\, \bar{n}\, .\ \overline{\phi[n]}
\end{array}$$

$$\overline{\mathsf{rec}(n,\ z,\ \lambda m u.\, s)} \;=\; \mathsf{rec}(\,\bar{n}\,,\ \bar{z}\,,\ \lambda m\,\bar{u}\,.\ \bar{s}\,)$$

For the variable $x : X$, the symbol $\bar{x}$ (either as a term or as a λ-binding) is not a translation as such, but a new variable of type $\overline{X}$. We shall also define

$$\overline{\mathsf{focus}\, P} \;=\; \mathsf{force}\ \overline{P}\,.$$

Notice that focus is *not* translated into focus, and we shall not allow terms to involve it in the main part of the discussion. The problem is that the translation $\overline{\Gamma} \vdash\ \overline{P} : \Sigma^2\, \overline{X}$ does not respect the equation that says that P is prime and so justifies use of the focus operator. However, from Lemma 8.13 and Proposition A 8.10, focus is only needed on the *outside* of a term, and then only for type $\mathbb{N}$, so we shall deal with it separately at the end of the section.

Notation 10.2 For any context Γ, define $\overline{\Gamma}$ by replacing each typed variable $x : X$ in Γ by $\bar{x} : \overline{X}$. Similarly, let $i_\Gamma : \Gamma \rightarrowtail \overline{\Gamma}$ be the list of $[i_X(x)/\,\bar{x}\,] : X \rightarrowtail \overline{X}$. We write i_X^* and i_Γ^* for the *substitutions* that relate terms involving these *free* variables, whereas Σ^{i_X} relates the corresponding λ-abstractions, *cf.* Lemma 10.7 below. Hence

$$i_{[\,]}^*\,\bar{a} \;=\; \bar{a} \qquad \text{and} \qquad i_{[\Gamma, x:X]}^*\,\bar{a} \;=\; [i_X(x)/\,\bar{x}\,]^* i_\Gamma^*\,\bar{a}\,.$$

Most of this section is devoted to proving

Theorem 10.3 For any term $\Gamma \vdash a : X$ in the comprehension calculus *without* focus, $\overline{\Gamma} \vdash\ \bar{a} : \overline{X}$ is well formed in the original calculus, whilst

$$\Gamma\ \vdash\ a \;=\; \mathsf{admit}_X(i_\Gamma^*\,\bar{a}\,) : X \qquad \text{and} \qquad \Gamma\ \vdash\ i_X a \;=\; i_\Gamma^*\,\bar{a}\ :\ \overline{X}$$

are provable in the comprehension calculus, *i.e.* the square

$$\begin{tikzcd}
\overline{\Gamma} \arrow[r, "\bar{a}"] & \overline{X} \arrow[d, shift left=2, "\mathsf{admit}_X"] \\
\Gamma \arrow[u, rightarrowtail, "i_\Gamma"] \arrow[r, "a"] & X \arrow[u, rightarrowtail, "i_X"]
\end{tikzcd}$$

commutes. The two forms are equivalent by the $\{\}\beta$ and $\{\}\eta$ rules, Lemmas 9.9 and 9.10.

In particular, if Γ and X don't involve comprehension then $\Gamma \vdash a = \bar{a} : X$.

The proof is by structural induction on the term $\Gamma \vdash a : X$, and therefore has a case for each connective: λ, i, admit, I, *etc.* Each of these *cases* is a lemma below, whereas the lemmas in the previous section were self-contained. To avoid repetitious language, the enunciation of each lemma is simply the introduction or elimination rule that defines the connective; by this we mean to assert that "if the premises (sub-terms) obey the equation $a = \mathsf{admit}_X(i_\Gamma^*\,\bar{a}\,)$ then so does the conclusion".

Since the transformation is also applied to nuclei (Lemma 10.10), the proof is actually on the *history* of formation of a, rather than simply on the expression without type annotations.

Variables, constants and algebraic operations

Lemma 10.4 For $x : X \vdash x : X$, we have $\mathsf{admit}_X(i_X^*\,\bar{x}\,) = x$.

PROOF. $i_X^*\,\bar{x}\, = [i_X x/\,\bar{x}\,]^*\,\bar{x}\, = i_X x$, and $\mathsf{admit}_X(i_X x) = x$ by Lemma 9.10 ($\{\}\eta$ for X). ∎

There is nothing to prove for the constants $\top$, $\bot$ and 0, and almost nothing for succ, so we just consider $\wedge$ as (an example of) an algebraic operation.

Lemma 10.5

$$\frac{\Gamma\ \vdash\ \phi : \Sigma^X \qquad \Gamma\ \vdash\ \psi : \Sigma^X}{\Gamma\ \vdash\ \phi\wedge\psi : \Sigma^X}$$

Recall from the statement of Theorem 10.3 that, by displaying this rule, we mean to claim that if ϕ and ψ satisfy a certain equation then so does $\phi \wedge \psi$.

PROOF. The general plan is to expand the term in the conclusion of the rule (here $\phi\wedge\psi$), using the induction hypothesis for the sub-terms (here ϕ and ψ) and the formulae for admit (Notation 9.2) and i_Γ^* (Notation 10.2), then apply the extended "rules" from the previous section, followed by the same procedure in reverse to obtain the translation of the term in question.

$$\begin{array}{rcll}
\phi\wedge\psi & = & (\mathsf{admit}_{\Sigma^X}\ i_\Gamma^*\,\overline{\phi}\,)\ \wedge\ (\mathsf{admit}_{\Sigma^X}\ i_\Gamma^*\,\overline{\psi}\,) & \text{induction hypothesis for } \phi,\,\psi \\
& = & (\Sigma^{i_X} i_\Gamma^*\,\overline{\phi}\,)\ \wedge\ (\Sigma^{i_X} i_\Gamma^*\,\overline{\psi}\,) & \text{Notation 9.2} \\
& = & \Sigma^{i_X}(i_\Gamma^*\,\overline{\phi}\,\wedge i_\Gamma^*\,\overline{\psi}\,) & \Sigma^{i_X}\ \text{is a homomorphism} \\
& = & \Sigma^{i_X} i_\Gamma^*(\,\overline{\phi}\,\wedge\,\overline{\psi}\,) & \text{subsitution into } \wedge \\
& = & \Sigma^{i_X} i_\Gamma^*\,\overline{\phi\wedge\psi} & \text{Notation 10.1} \\
& = & \mathsf{admit}_{\Sigma_X}\ i_\Gamma^*\,\overline{\phi\wedge\psi} & \text{Notation 9.2.}\ \blacksquare
\end{array}$$

Lemma 10.6

$$\frac{\Gamma,\ n:\mathbb{N}\ \vdash\ \phi[n] : \Sigma^X}{\Gamma\ \vdash\ \exists n.\,\phi[n] : \Sigma^X}$$

PROOF.

$$\begin{array}{rcll}
\exists n.\,\phi[n] & = & \exists n.\ \mathsf{admit}_{\Sigma^X}\ i_{\Gamma,n:\mathbb{N}}^*\,\overline{\phi[n]} & \text{induction hypothesis for } \phi[n] \\
& = & \exists n.\,\Sigma^{i_X} i_\mathbb{N}^* i_\Gamma^*\,\overline{\phi[n]} & \text{Notation 9.2 and 10.2} \\
& = & \Sigma^{i_X}\exists n.\ i_\mathbb{N}^* i_\Gamma^*\,\overline{\phi[n]} & \Sigma^{i_X}\ \text{is a homomorphism} \\
& = & \Sigma^{i_X} i_\Gamma^*\exists\,\bar{n}\,.\ \overline{\phi[n]} & \text{substitution into } \exists \\
& = & \mathsf{admit}_{\Sigma^X}\ i_\Gamma^*\,\overline{\exists n.\,\phi[n]} & \text{Notation 9.2 and 10.1.}\ \blacksquare
\end{array}$$

The lambda calculus

The difference between Σ^{i_Y} and i_Y^* is that Σ^{i_Y} acts on the exponent of the type by λ-reduction, whilst i_Y^* acts on the context by substitution.

Lemma 10.7 If $\Gamma,\ \bar{y} : \overline{Y} \vdash \theta : \Sigma^{\overline{Z}}$ then $\Gamma \vdash \Sigma^{i_Y \times i_Z} \lambda\, \bar{y}\,.\, \theta = \lambda y.\, \Sigma^{i_Z} i_Y^* \theta : \Sigma^{Y \times Z}$.

PROOF.

$$\begin{aligned}
\Sigma^{i_Y \times i_Z} \lambda\, \bar{y}\,.\, \theta &= \lambda yz.\, (\lambda\, \bar{y}\,.\, \theta)(i_Y y)(i_Z z) && \text{definition of } \Sigma^{i_Y \times i_Z} \\
&= \lambda yz.\, \big([i_Y(y)/\,\bar{y}\,]^* \theta\big)(i_Z z) && \lambda\beta \\
&= \lambda yz.\, (i_Y^* \theta)(i_Z z) && \text{definition of } i_Y^* \\
&= \lambda y.\, \Sigma^{i_Z} i_Y^* \theta && \text{definition of } \Sigma^{i_Z}.\ \blacksquare
\end{aligned}$$

Lemma 10.8

$$\frac{\Gamma,\ y : Y\ \vdash\ \phi : \Sigma^Z}{\Gamma\ \vdash\ \lambda y.\, \phi : \Sigma^{Y \times Z}}$$

PROOF. We use Lemma 10.7 with $\Gamma,\ \bar{y} : \overline{Y} \vdash \theta \ \equiv\ i_\Gamma^* \,\overline{\phi} : \Sigma^{\overline{Z}}$.

$$\begin{aligned}
\lambda y.\, \phi &= \lambda y.\, \mathsf{admit}_{\Sigma^Z}\, i_{\Gamma, y:Y}^* \,\overline{\phi} && \text{induction hypothesis for } \phi \\
&= \lambda y.\, \Sigma^{i_Z} i_Y^* i_\Gamma^* \,\overline{\phi} && \text{Notation 9.2 and 10.2} \\
&= \Sigma^{i_Y \times i_Z} \lambda\, \bar{y}\,.\, i_\Gamma^* \,\overline{\phi} && \text{Lemma 10.7} \\
&= \Sigma^{i_Y \times i_Z} (i_\Gamma^* \lambda\, \bar{y}\,.\, \overline{\phi}\,) && \text{substitution into } \lambda \\
&= \mathsf{admit}_{\Sigma^{Y \times Z}}\, i_\Gamma^* \,\overline{\lambda y.\, \phi} && \text{Notation 9.2 and 10.1.}\ \blacksquare
\end{aligned}$$

Lemma 10.9

$$\frac{\Gamma\ \vdash\ \phi : \Sigma^{Y \times Z} \qquad \Gamma\ \vdash\ a : Y}{\Gamma\ \vdash\ \phi[a] : \Sigma^Z}$$

PROOF.

$$\begin{aligned}
\phi a &= (\mathsf{admit}_{\Sigma^{Y \times Z}}\, i_\Gamma^* \,\overline{\phi}\,)(\mathsf{admit}_Y\, i_\Gamma^* \,\bar{a}\,) && \text{induction hypothesis for } \phi \text{ and } a \\
&= (\Sigma^{Y \times i_Z} \Sigma^{i_Y \times \overline{Z}}\, i_\Gamma^* \,\overline{\phi}\,)(\mathsf{admit}_Y\, i_\Gamma^* \,\bar{a}\,) && \text{Notation 9.2} \\
&= \Sigma^{i_Z} \big(i_\Gamma^* \,\overline{\phi}\, (i_Y\, \mathsf{admit}_Y\, i_\Gamma^* \,\bar{a}\,)\big) \\
&= \Sigma^{i_Z} \big(i_\Gamma^* \,\overline{\phi}\, (i_\Gamma^* \,\bar{a}\,)\big) && \{\}\beta \text{ for } Y \\
&= \Sigma^{i_Z} i_\Gamma^* (\,\overline{\phi}\ \bar{a}\,) && \text{substitution into } \overline{\phi}\ \bar{a} \\
&= \mathsf{admit}_{\Sigma^Z}\, i_\Gamma^* \,\overline{\phi a} && \text{Notation 9.2 and 10.1.}\ \blacksquare
\end{aligned}$$

Subtypes

Here is where we use the main induction hypothesis in Theorem 10.3 for the nucleus E that defines the subtype $\{X \mid E\}$. The prohibition on focus is not a problem: it can be eliminated, as the type of E is Σ^X (Proposition A 8.10).

Lemma 10.10 $\Sigma^{i_X} \cdot \overline{E} = E \cdot \Sigma^{i_X}$ and $E_{\{X|E\}} = I_X \cdot E \cdot \Sigma^{i_X} = \overline{E}\,$.

PROOF. The induction hypothesis for $E\phi$ and ϕ yields

$$E\phi\ =\ \mathsf{admit}_{\Sigma^X}(\,\overline{E}\ \overline{\phi}\,)\ =\ \Sigma^{i_X}(\,\overline{E}\ \overline{\phi}\,) \quad \text{and} \quad \phi\ =\ \mathsf{admit}_{\Sigma^X}\ \overline{\phi}\ \ =\ \Sigma^{i_X}\,\overline{\phi},$$

so $E(\Sigma^{i_X}\,\overline{\phi}\,)\ =\ \Sigma^{i_X}(\,\overline{E}\ \overline{\phi}\,)$, which is the first equation. The induction hypothesis also says that $E_X = \overline{E_X}\,$, since the type of E_X doesn't involve comprehension. Then

$$E_{\{X|E\}}\ =\ I_X \cdot E \cdot \Sigma^{i_X}\ =\ I_X \cdot \Sigma^{i_X} \cdot \overline{E}\ =\ E_X \cdot \overline{E}\ =\ \overline{E_X \cdot E}\ =\ E$$

by Notation 10.2 and Lemmas 9.4, 10.9 and 9.8. ■

Lemma 10.11

$$\frac{\Gamma \ \vdash \ a : X \qquad \Gamma,\ \phi : \Sigma^X \ \vdash \ \phi a = E\phi a}{\Gamma \ \vdash \ \mathsf{admit}_{X,E}\, a : \{X \mid E\}}$$

PROOF.

$$\begin{aligned}
\mathsf{admit}_{X,E}\, a &= \mathsf{admit}_{X,E}\, \mathsf{admit}_X\, i_\Gamma^*\, \bar{a} && \text{induction hypothesis for } a\\
&= \mathsf{admit}_{\{X|E\}}\, i_\Gamma^*\, \bar{a} && \text{Notation 9.2}\\
&= \mathsf{admit}_{\{X|E\}}\, i_\Gamma^*\, \overline{\mathsf{admit}_{X,E}\, a} && \text{Notation 10.2.} \blacksquare
\end{aligned}$$

Lemma 10.12

$$\frac{\Gamma \ \vdash \ a : \{X \mid E\}}{\Gamma \ \vdash \ i_{X,E} a : X}$$

PROOF.

$$\begin{aligned}
i_{X,E} a &= i_{X,E}\, \mathsf{admit}_{\{X|E\}}\, i_\Gamma^*\, \bar{a} && \text{induction hypothesis for } a\\
&= i_{X,E}\, \mathsf{admit}_{X,E}\, \mathsf{admit}_X\, i_\Gamma^*\, \bar{a} && \text{Notation 9.2}\\
&= \mathsf{admit}_X\, i_\Gamma^*\, \bar{a} && \{\}\eta \text{ for } X, E\\
&= \mathsf{admit}_X\, i_\Gamma^*\, \overline{i_{X,E} a}\,. && \text{Notation 10.2.} \blacksquare
\end{aligned}$$

Lemma 10.13

$$\frac{\Gamma \ \vdash \ \theta : \Sigma^{\{X|E\}}}{\Gamma \ \vdash \ I_{X,E}\theta : \Sigma^X}$$

PROOF.

$$\begin{aligned}
I_{X,E}\theta &= I_{X,E}\, \mathsf{admit}_{\Sigma^{\{X|E\}}}\, i_\Gamma^*\, \bar{\theta} && \text{induction hypothesis for } \theta\\
&= I_{X,E}\Sigma^{i_{X,E}}\Sigma^{i_X} i_\Gamma^*\, \bar{\theta} && \text{Notation 9.2}\\
&= E\Sigma^{i_X} i_\Gamma^*\, \bar{\theta} && \Sigma^{\{\}}\beta \text{ for } X, E\\
&= \Sigma^{i_X}\ \overline{E}\, i_\Gamma^*\, \bar{\theta} && \text{Lemma 10.10}\\
&= \Sigma^{i_X} i_\Gamma^*\ \overline{E}\ \bar{\theta} && \text{substitution into } \overline{E}\ \bar{\theta}\\
&= \mathsf{admit}_{\Sigma^X}\ i_\Gamma^*\ \overline{I_{X,E}\theta} && \text{Notation 9.2 and 10.2.} \blacksquare
\end{aligned}$$

Recursion

Lemma 10.14

$$\frac{\Gamma \ \vdash \ n : \mathbb{N} \qquad \Gamma \ \vdash \ z : X \qquad \Gamma,\ m : \mathbb{N},\ x : X \ \vdash \ s(m,x) : X}{\Gamma \ \vdash \ \mathsf{rec}(n,\ z,\ \lambda mx.\, s) : X}$$

PROOF. Let $\Gamma \vdash z' = i_\Gamma^*\, \bar{z} \,:\, \overline{X}$ and $\Gamma,\ m : \mathbb{N},\ \bar{x} \,:\, \overline{X} \vdash s'(m,\ \bar{x}\,) = i_\Gamma^*\, \bar{s}\,(m,\ \bar{x}\,) :\ \overline{X}$.

So the induction hypotheses say that $\Gamma \vdash n = i_\Gamma^*\, \bar{n}\ : \mathbb{N}$, $\quad \Gamma \vdash z = \mathsf{admit}_X\, z' : X$ and

$$\Gamma,\ m : \mathbb{N},\ x : X \ \vdash \ s(m,x) = \mathsf{admit}_X\, s'(m, i_X x) : X.$$

We use Lemma 8.14, or rather its analogue with the extended rules of Section 9 in place of the $\{\}\beta$- and η-rules; the symbols $\bar{x}$, z' and s' here correspond to y, z and s there.

$$
\begin{array}{rlr}
\mathsf{rec}(n,\ z,\ \lambda mx.\, s) & & \\
= & \mathsf{rec}\big(n,\ \mathsf{admit}_X\, z',\ \lambda mx.\ \mathsf{admit}_X\, s'(m, i_X x)\big) & \text{induction hypothesis} \\
= & \mathsf{admit}_X\, \mathsf{rec}\big(n,\ z',\ \lambda m\,\bar{x}\,.\, s'(m,\bar{x}\,)\big) & \text{Lemma 8.14} \\
= & \mathsf{admit}_X\, \mathsf{rec}\big(i_\Gamma^*\,\bar{n}\,,\ i_\Gamma^*\,\bar{z}\,,\ \lambda m\,\bar{x}\,.\, i_\Gamma^*\,\bar{s}\,(m,\bar{x}\,)\big) & \text{definition} \\
= & \mathsf{admit}_X\, i_\Gamma^* \mathsf{rec}\big(\,\bar{n}\,,\ \ \bar{z}\,,\ \lambda m\,\bar{x}\,.\ \bar{s}\,(m,\bar{x}\,)\big) & \text{substitution into } \mathsf{rec} \\
= & \mathsf{admit}_X\, i_\Gamma^*\, \overline{\mathsf{rec}(n,\ z,\ \lambda mx.\, s)}\, : X & \text{Notation 10.2.} \ \blacksquare
\end{array}
$$

Sobriety

We have now completed the proof of Theorem 10.3, that, for any term $\Gamma \vdash a : X$ in the comprehension calculus *without* focus,

$$\Gamma\ \vdash\ a = \mathsf{admit}_X(i_\Gamma^*\,\bar{a}\,) : X \qquad \text{and} \qquad \Gamma\ \vdash\ i_X a = i_\Gamma^*\,\bar{a}\ :\ \overline{X}\,.$$

There are two ways of handling a term $\mathsf{focus}\, P : \mathbb{N}$.

Remark 10.15 Lemma 8.11 says that $\mathbb{N} \cong \{\Sigma^2\mathbb{N} \mid E_{\mathbb{N}}\}$, where $E_{\mathbb{N}}$ characterises primes. This makes use of admit instead of focus. More generally, $\{\mathbb{N} \mid E\} \cong \{\Sigma^2\mathbb{N} \mid E'\}$, where E' encodes the composite inclusion. Another way of saying this is that, in Notation 9.2, we define the base case $\overline{\mathbb{N}}$ as $\Sigma^2\mathbb{N}$ instead of $\mathbb{N}$. This way we only ever deal with subspaces of injectives (*cf.* Corollary 5.3), and never need to use focus.

Remark 10.16 Alternatively, if $\Gamma \vdash P : \Sigma^2 X$ is a prime not involving focus,

$$
\begin{array}{rclr}
\mathsf{focus}_X\, P & = & \mathsf{admit}_X\, \big(\mathsf{focus}_{\overline{X}}\, (\Sigma^2 i_X P)\big) & \text{Lemma 8.13} \\
& = & \mathsf{admit}_X\, \big(\mathsf{focus}_{\overline{X}}\, (i_{\Sigma^2 X} P)\big) & \text{Notation 9.2} \\
& = & \mathsf{admit}_X\, \big(\mathsf{focus}_{\overline{X}}\, (i_\Gamma^*\, \overline{P}\,)\big) & \text{Theorem 10.3} \\
& = & \mathsf{admit}_X\, (i_\Gamma^*\, \mathsf{force}_{\overline{X}}\ \ \overline{P}\,) & \textit{cf.}\ \text{Lemma A 8.4} \\
& = & \mathsf{admit}_X\, (i_\Gamma^*\, \overline{\mathsf{focus}_X\, P}\,) & \text{Notation 10.2.}
\end{array}
$$

Lemma A 8.4 actually said

$$\frac{u : \Delta \to \Gamma \qquad \Gamma\ \vdash\ P : \Sigma^2 X \quad \text{prime}}{\Delta\ \vdash\ u^* P : \Sigma^2 X \quad \text{prime}}$$

$$\Delta\ \vdash\ u^*(\mathsf{focus}\, P)\ =\ \mathsf{focus}(u^* P) : X$$

and we're trying to use its converse, which is not valid. Although $\Gamma \vdash i_\Gamma^*\, \overline{P}\ :\ \Sigma^2\, \overline{X}$ is prime, $\overline{\Gamma} \vdash\ \overline{P}\ : \Sigma^2\, \overline{X}$ need not be. This is why we write force instead of focus here.

Semantically, $\Gamma \rightarrowtail \overline{\Gamma}$ is a subspace, on which the relevant equation holds, but it need not hold on the ambient space $\overline{\Gamma}$. That such a situation arises is more obvious for descriptions (Remark A 9.13): *any* predicate $\overline{\Gamma} \vdash \phi : \Sigma^{\mathbb{N}}$ becomes a description when restricted to the locally closed subspace $\Gamma \rightarrowtail \overline{\Gamma}$ defined by

$$(\exists n.\, \phi[n])\ =\ \top \quad \text{and} \quad (\exists nm.\, \phi[n] \wedge \phi[m] \wedge n \neq m)\ =\ \bot.$$

In the terms of this paper, the equation in Definition 4.5 defining a (prime or) homomorphism on a subspace $\{Y_1 \mid E_1\}$ involves $\widehat{E_1}$, and does not imply the form without it.

Nevertheless, it is a legitimate alternative way of treating $\mathsf{focus}\, P$ to translate it into $\mathsf{force}\ \overline{P}$. For, if $\Gamma \vdash \psi : \Sigma^X$ then

$$\psi[\mathsf{focus}\, P]\ =\ P\psi\ =\ (\mathsf{admit}_{\Sigma^2 X}\, i_\Gamma^*\, \overline{P}\,)(\mathsf{admit}_{\Sigma^X}\, i_\Gamma^*\, \overline{\psi}\,)\ =\ i_\Gamma^*\, \overline{P\psi}\ \ =\ i_\Gamma^*\, \overline{\psi}\, [\mathsf{force}\ \overline{P}\,]$$

using the unrestricted β-rule for force.

As we know from Sections A 8 and 11, and the work on computational effects cited there, we may obtain *different* terms $\overline{\Gamma} \vdash \bar{a} : \overline{X}$ by applying or reversing the β- and η-rules of the λ-calculus with force. So, for such a calculus to be defined consistently, an order of evaluation must be specified.

However, these computational terms with possibly different denotations on $\overline{\Gamma}$ become *equal* when we apply i_Γ^* to restrict them to the required context Γ. In other words, as we know very well from experience, there are different programs $\overline{\Gamma} \vdash \bar{a} : \overline{X}$, possibly involving computational effects, to compute the same denotational value $\Gamma \vdash a : X$.

Categorical equivalence

Theorem 10.17 Let $\mathcal{C}$ and $\mathcal{D}$ be the categories generated by the restricted λ-calculus (possibly with the extra structure) and the comprehension calculus. Then there is an equivalence of categories $\overline{\mathcal{C}} \simeq \mathcal{D}$ whose effect on types is

$$\begin{tikzcd}
\{X \mid E\} \arrow[r, mapsto] & \{X \mid E\} \\
\overline{\mathcal{C}} \arrow[r, shift left, "\simeq"'] & \mathcal{D} \arrow[l, shift left] \\
\{\,\overline{X} \mid \overline{E_X}\,\} & X, \arrow[l, mapsto]
\end{tikzcd}$$

where the notation $\{X \mid E\}$ in $\overline{\mathcal{C}}$ is the categorical one in Section 4, whilst in $\mathcal{D}$ it is understood as the comprehension calculus in Section 8.

Proof. The previous section defined the transformation $\mathcal{D} \to \overline{\mathcal{C}}$ on types, and showed that the composite $\mathcal{D} \to \overline{\mathcal{C}} \to \mathcal{D}$ is isomorphic to the identity: Proposition 9.11 provided mutually inverse terms in the comprehension calculus that relate any type X to $\{\,\overline{X} \mid E_X\}$. The other composite *is* the identity on types, so we just have to show that $\overline{\mathcal{C}} \to \mathcal{D}$ is full and faithful.

Let X_1 and X_2 be types that are defined without comprehension, and E_1 and E_2 nuclei on them. We shall show that there is a bijection between the $\overline{\mathcal{C}}$-morphisms

$$\widehat{J} : \{X_1 \mid E_1\} \longrightarrow \{X_2 \mid E_2\}$$

defined in Section 4 and the terms

$$x : \{X_1 \mid E_1\} \ \vdash\ \mathsf{focus}\, P : \{X_2 \mid E_2\}$$

that we have just expressed in normal form. Of course $\overline{\{X_1 \mid E_1\}} = X_1$, and we write $i_1 = i_{X_1}$, *etc.*

In each of the categories, these morphisms correspond bijectively to homomorphisms

$$\Sigma^{\{X_2|E_2\}} \longrightarrow \Sigma^{\{X_1|E_1\}},$$

but first we consider ordinary maps, without the homomorphism requirement.

By Proposition 6.2 for $\overline{\mathcal{C}}$, $\Sigma^{\{X|E\}} \cong \{\Sigma^X \mid \Sigma^E\}$, which is simply a retract of the power type Σ^X in $\mathcal{C}$ (the restricted λ-calculus), so a $\overline{\mathcal{C}}$-morphism $J : \Sigma^{\{X_2|E_2\}} \to \Sigma^{\{X_1|E_1\}}$ is a $\mathcal{C}$-morphism $J : \Sigma^{X_2} \to \Sigma^{X_1}$ such that $E_2 \,;\, J = J = J \,;\, E_1$.

$$\begin{tikzcd}
\Sigma^{X_2} \arrow[r, "J"] \arrow[d, shift left, "E_2"] & \Sigma^{X_1} \arrow[d, shift left, "E_1"] \\
\Sigma^{\{X_2|E_2\}} \arrow[r, "J"] \arrow[u, shift left, "E_2"] & \Sigma^{\{X_1|E_1\}} \arrow[u, shift left, "E_1"]
\end{tikzcd}
\qquad
\begin{tikzcd}
\Sigma^{X_2} \arrow[r, "\overline{H}"] \arrow[d, shift left, "\Sigma^{i_2}"] & \Sigma^{X_1} \arrow[d, shift left, "\Sigma^{i_1} = \mathsf{admit}_{\Sigma^{X_1}}"] \\
\Sigma^{\{X_2|E_2\}} \arrow[r, "H"] \arrow[u, shift left, "i_{\Sigma^{X_2}} = I_2"] & \Sigma^{\{X_1|E_1\}} \arrow[u, shift left, "I_1"]
\end{tikzcd}$$

The corresponding power type in $\mathcal{D}$ is also a retract, and Theorem 10.3 characterised its morphisms H in the same fashion. Hence $\mathsf{H}\overline{\mathcal{C}} \simeq \mathsf{H}\mathcal{D}$.

Now, homomorphisms are characterised equationally amongst all maps, and we have already shown that the functor is full and faithful for them, *i.e.* that maps and their equations agree, so the homomorphisms also agree.

In $\overline{\mathcal{C}}$, the equation that defines homomorphisms is more clearly understood by means of Lemma 6.6, *i.e.* with reference to the subtypes $\{X_i \mid E_i\}$, rather than using Definition 4.5, which transfers the condition to the ambient types X_i. We have seen the same phenomenon in $\mathcal{D}$, where the translation $\overline{P}$ of a prime P relative to the sub-types need not be prime with respect to the translated types.

That deals with single-type contexts, but that is enough as products may be encoded as comprehension types in each category. ■

Therefore our first construction, adjoining formal Σ-split equalisers to the category in Sections 4–6, is equivalent to the third, the extension of the λ-calculus in Sections 8–10.

11 Sums and quotients

Paré's theorem, that any elementary topos has the monadic property [Par74], which originally inspired abstract Stone duality, was itself motivated by the simple way that it affords for constructing colimits. So we conclude this paper with applications of the comprehension calculus to coproducts and coequalisers.

Example 11.1 The idea of Stone duality is to consider spaces in terms of the corresponding algebras. For any algebra (A, α), we have

$$\mathsf{pts}(A,\alpha) \;=\; \{\Sigma^A \mid \eta_A \cdot \alpha \equiv \lambda F\phi.\, \phi(\alpha F)\},$$

in which $\eta_A \cdot \alpha$ is a nucleus (Definition 8.7) because of the Eilenberg–Moore equations for α. The homomorphism $H : B \to A$ corresponds to the function

$$x : \mathsf{pts}(A,\alpha) \;\vdash\; \mathsf{admit}\big(\lambda\psi.\, H\psi(ix)\big) : \mathsf{pts}(B,\beta),$$

in which H is behaving as a continuation-transformer. (We saw in the previous section that the operators i and admit merely serve as compile-time type-annotation.) In particular, if $H = \Sigma^f$, where $f : X \to Y$, this is just $\mathsf{focus}\big(\lambda\psi.\, \psi(fx)\big) = fx$ by Lemma 8.11. ■

The continuation-passing style is more clearly visible in a more complicated example.

Example 11.2 The coproduct $X + Y$ of spaces corresponds to the product $A = \Sigma^X \times \Sigma^Y$ of algebras, whose structure map $\alpha = \langle P_0, P_1\rangle$ was given in Lemma 5.5:

$$P_0 : \Sigma^2(\Sigma^X \times \Sigma^Y) \to \Sigma^X \qquad \text{by} \quad \mathcal{H} \mapsto \lambda x.\, \mathcal{H}\,(\lambda\phi\psi.\, \phi x).$$

Then $X + Y = \mathsf{pts}(A, \alpha) = \{\Sigma^{\Sigma^X \times \Sigma^Y} \mid E_{X+Y}\}$, where

$$E_{X+Y} \;=\; \eta_{\Sigma^X \times \Sigma^Y} \cdot \alpha : \mathcal{H} \mapsto \lambda H.\, H\,\langle P_0\mathcal{H}, P_1\mathcal{H}\rangle.$$

The inclusion $\nu_0 : X \to X + Y$ satisfies $\Sigma^{\nu_0} = \pi_0$, so

$$\nu_0(x) = \mathsf{admit}(\lambda\phi\psi.\, \phi x) \quad \text{and} \quad \nu_1(y) = \mathsf{admit}(\lambda\phi\psi.\, \psi y).$$

Then, for $f : X \to Z$ and $g : Y \to Z$, the mediator $X + Y \to Z$ is

$$[f, g] : u \mapsto \mathsf{focus}\big(\lambda\theta.\, (iu)\big(\lambda x.\, \theta(fx)\big)\big(\lambda y.\, \theta(gy)\big)\big),$$

in which the continuation θ from Z is passed either as $\theta \cdot f$ to X or as $\theta \cdot g$ to Y. When $u = \nu_0(x)$ or $\nu_1(y)$, one of the two branches is selected and that continuation is applied to x or y. ■

We already know that *products* distribute over coproducts (Proposition A 7.11), and coproducts are disjoint and stable under *pullback* on the assumption that Σ is a distributive lattice and satisfies the Euclidean principle (Section C 9). In fact this extra structure is unnecessary: if $(\mathcal{C}, \Sigma)$ is monadic then $\mathcal{C}$ is extensive on a very much weaker assumption.

Lemma 11.3 The map $\mathbf{0} \to \mathbf{1}$ is a Σ-split mono iff Σ has a constant.

PROOF. $I : \Sigma^{\mathbf{0}} = \mathbf{1} \to \Sigma = \Sigma^{\mathbf{1}}$. ■

If $I = \top$, this makes $\mathbf{0}$ a *closed* subspace; if $I = \bot$ then it's open (Examples 2.5). However, we shall call the constant $\star$ here to emphasise that we are *not* using any lattice structure. Then coproduct inclusions are also Σ-split monos:

Lemma 11.4 $X \cong \{X + Y \mid \langle \pi_0, \star \rangle\}$ and $Y \cong \{X + Y \mid \langle \star, \pi_1 \rangle\}$.

PROOF. These idempotents arise from the diagram

$$\begin{tikzcd} \Sigma^X \arrow[r, shift left, "\langle \mathsf{id}, \star \rangle"'] & \Sigma^X \times \Sigma^Y \arrow[l, shift left, two heads, "\pi_0 \equiv \Sigma^{\nu_0}"'] \arrow[r, shift left, two heads, "\pi_1 \equiv \Sigma^{\nu_1}"] & \Sigma^Y \arrow[l, shift left, "\langle \star, \mathsf{id} \rangle"] \end{tikzcd}$$

■

We want to show that every map $U \to \mathbf{2}$ arises in this way from a unique coproduct. Because of the continuation-passing behaviour, it is convenient to treat $\mathbf{2}$ as a subspace of $\Sigma^{\Sigma \times \Sigma}$. Then the definable elements are $\pi_0 \equiv \lambda xy.\, x$ and $\pi_1 \equiv \lambda xy.\, y$, which simply select the appropriate continuation from the pair.

Lemma 11.5 $\mathbf{2} \cong \{\Sigma^{\Sigma \times \Sigma} \mid \lambda \mathcal{F} F.\, F(\mathcal{F}\pi_0, \mathcal{F}\pi_1)\}$.

PROOF. This is the primality equation, $\mathcal{F}P = P\big(\lambda x.\, \mathcal{F}(\lambda \phi.\, \phi x)\big)$, with λx replaced by the two values $0, 1 : \mathbf{2}$, and $\phi : \Sigma^{\mathbf{2}}$ by a pair. ■

We shall abuse our language by calling the two-argument functions $P : \Sigma^{\Sigma \times \Sigma}$ that belong to this subspace "prime". However, we need yet another characterisation of them.

Lemma 11.6 $\Gamma \vdash P : \Sigma^{\Sigma \times \Sigma}$ is prime iff, for $x, y, z : \Sigma$ and $\mathcal{G} : \Sigma^{\Sigma^\Sigma}$,

$$P(x,x) \;=\; x \qquad\qquad P\big(P(x,y), z\big) \;=\; P(x,z) \;=\; P\big(x, P(y,z)\big)$$

$$\mathcal{G}\big(\lambda z.\, P(z,\star)\big) \;=\; P\big(\mathcal{G}\mathsf{id},\, \mathcal{G}(\lambda z.\, \star)\big) \qquad \mathcal{G}\big(\lambda z.\, P(\star, z)\big) \;=\; P\big(\mathcal{G}(\lambda z.\, \star),\, \mathcal{G}\mathsf{id}\big).$$

PROOF. If P is prime, so $\Gamma,\ \mathcal{F} : \Sigma^2(\Sigma \times \Sigma) \;\vdash\; \mathcal{F}P \;=\; P(\mathcal{F}\pi_0, \mathcal{F}\pi_1)$, then

- put $\mathcal{F} \equiv \lambda Q.\, x$, so $\mathcal{F}P = \mathcal{F}\pi_0 = \mathcal{F}\pi_1 = x$ and primality says that $x = P(x,x)$;
- put $\mathcal{F} \equiv \lambda Q.\, Q\big(Q(x,y), z\big)$, so $\mathcal{F}\pi_0 = x$, $\mathcal{F}\pi_1 = z$ and $P\big(P(x,y), z\big) \equiv \mathcal{F}P = P(x,z)$;
- put $\mathcal{F} \equiv \lambda F.\, \mathcal{G}\big(\lambda z.\, F(z,\star)\big)$, so $\mathcal{F}\pi_0 = \mathcal{G}\mathsf{id}$, $\mathcal{F}\pi_1 = \mathcal{G}(\lambda z.\, \star)$ and primality says that $\mathcal{G}\big(\lambda z.\, P(z,\star)\big) \equiv \mathcal{F}P = P\big(\mathcal{G}\mathsf{id},\, \mathcal{G}(\lambda z.\, \star)\big)$.

Conversely, we first put $\mathcal{G} \equiv \lambda\theta.\, \mathcal{F}\big(\lambda xy.\, \theta P(x,y)\big)$, so $\mathcal{G}(\lambda z.\, \star) = \mathcal{F}(\lambda xy.\, \star)$, $\mathcal{G}\mathsf{id} = \mathcal{F}P$,

$$\mathcal{G}\big(\lambda z.\, P(z,\star)\big) \;=\; \mathcal{F}\big(\lambda xy.\, P(P(x,y), \star)\big) \;=\; \mathcal{F}\big(\lambda xy.\, P(x,\star)\big),$$

$$\mathcal{G}\big(\lambda z.\, P(\star, z)\big) \;=\; \mathcal{F}\big(\lambda xy.\, P(\star, P(x,y))\big) \;=\; \mathcal{F}\big(\lambda xy.\, P(\star, y)\big),$$

with which the equations involving $\mathcal{G}$ give

$$\mathcal{F}\big(\lambda xy.\, P(x,\star)\big) \;=\; P\big(\mathcal{F}P,\, \mathcal{F}(\lambda xy.\, \star)\big) \qquad \mathcal{F}\big(\lambda xy.\, P(\star, y)\big) \;=\; P\big(\mathcal{F}(\lambda xy.\, \star),\, \mathcal{F}P\big).$$

Next put $\mathcal{G} \equiv \lambda\theta.\,\mathcal{F}(\lambda xy.\,\theta x)$, so

$$\mathcal{G}(\lambda z.\,\star) \;=\; \mathcal{F}(\lambda xy.\,\star), \qquad \mathcal{G}\big(\lambda z.\,P(z,\star)\big) \;=\; \mathcal{F}\big(\lambda xy.\,P(x,\star)\big) \quad\text{and}\quad \mathcal{G}\mathsf{id} \;=\; \mathcal{F}\pi_0,$$

with which the first $\mathcal{G}$-equation gives

$$\mathcal{F}\big(\lambda xy.\,P(x,\star)\big) \;=\; P\big(\mathcal{F}\pi_0,\,\mathcal{F}(\lambda xy.\,\star)\big),$$

and similarly $\mathcal{G} \equiv \lambda\theta.\,\mathcal{F}(\lambda xy.\,\theta y)$ yields $\mathcal{F}\big(\lambda xy.\,P(\star,y)\big) = P\big(\mathcal{F}(\lambda xy.\,\star),\,F\pi_1\big)$.

From these four consequences of the higher-type $\mathcal{G}$-equations, together with the ones of base type, we deduce that P is prime:

$$\begin{aligned}
\mathcal{F}P &= P\big(P(\mathcal{F}P,\,\mathcal{F}(\lambda xy.\,\star)),\;P(\mathcal{F}(\lambda xy.\,\star),\mathcal{F}P)\big)\\
&= P\big(\mathcal{F}(\lambda xy.\,P(x,\star)),\qquad \mathcal{F}(\lambda xy.\,P(\star,y))\big)\\
&= P\big(P(\mathcal{F}\pi_0,\,\mathcal{F}(\lambda xy.\,\star)),\;P(\mathcal{F}(\lambda xy.\,\star),\,\mathcal{F}\pi_1)\big)\\
&= P(\mathcal{F}\pi_0,\mathcal{F}\pi_1).
\end{aligned}$$

■

Proposition 11.7 Let $u : U \vdash Pu : \Sigma^{\Sigma\times\Sigma}$ be prime. Then the following subspaces are well defined, their coproduct is U and the squares are pullbacks.

$$\begin{tikzcd}
\{U \mid E_0 \equiv \lambda\phi u.\,Pu(\phi u,\star)\} \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & U \arrow[d, "u \mapsto \mathsf{focus}(Pu)"] & \{U \mid E_1 \equiv \lambda\psi u.\,Pu(\star,\psi u)\} \arrow[l] \arrow[d] \arrow[dl, phantom, "\llcorner", very near start] \\
\mathbf{1} \arrow[r] & \mathbf{2} & \mathbf{1} \arrow[l]
\end{tikzcd}$$

Proof. To show that E_0 is a nucleus, we must show for $u : U$ and $\mathcal{F} : \Sigma^3 U$ that

$$E_0\big(\lambda v.\,\mathcal{F}(\lambda\phi.\,E_0\phi v)\big)u \;=\; Pu\big(\mathcal{F}(\lambda\phi.\,Pu(\phi u,\star)),\,\star\big) \;\equiv\; Pu(y,\star)$$

is equal to

$$E_0\big(\lambda v.\,\mathcal{F}(\lambda\phi.\,\phi v)\big)u \;=\; Pu\big(\mathcal{F}(\lambda\phi.\,\phi u),\,\star\big) \;\equiv\; Pu(x,\star).$$

Put $\mathcal{G} \equiv \lambda\theta.\,\mathcal{F}\big(\lambda\phi.\,\theta(\phi u)\big)$ in the Lemma. Then

$$\begin{aligned}
y &= \mathcal{F}\big(\lambda\phi.\,Pu(\phi u,\star)\big)\\
&= \mathcal{G}\big(\lambda z.\,Pu(z,\star)\big)\\
&= Pu\big(\mathcal{G}\mathsf{id},\mathcal{G}(\lambda z.\,\star)\big)\\
&= Pu\big(\mathcal{F}(\lambda\phi.\,\phi u),\,\mathcal{F}(\lambda\phi.\,\star)\big)\\
&\equiv Pu(x,t),
\end{aligned}$$

so $Pu(y,\star) = Pu\big(Pu(x,t),\star\big) = Pu(x,\star)$ as required.

To test that the left hand square is a pullback, we are given $\Gamma \vdash v : U$ such that $\Gamma \vdash Pv = \pi_0$. Then

$$\Gamma,\;\theta : \Sigma^U \;\vdash\; E_0\theta v \;=\; Pv(\theta v,\star) \;=\; \pi_0(\theta v,\star) \;=\; \theta v,$$

as is required to form the mediator $\Gamma \vdash \mathsf{admit}\,v : \{U \mid E_0\}$.

To show that U is the coproduct, we must show that Σ^U is the associated product, *i.e.* that $\theta : \Sigma^U$ corresponds bijectively to $\langle\phi,\psi\rangle$ where $\phi = E_0\phi$ and $\psi = E_1\psi$. In fact, P serves for the pairing operation as well as for the projections: $\theta = \lambda u.\,Pu(\phi u,\psi u)$.

$$\theta \;\mapsto\; \big\langle\lambda u.\,Pu(\theta u,\star),\,\lambda u.\,Pu(\star,\theta u)\big\rangle \;\mapsto\; \lambda u.\,Pu\big(Pu(\theta u,\star),\,Pu(\star,\theta u)\big) \;=\; \lambda u.\,\theta u$$

$$(\phi,\psi) \;\mapsto\; \lambda u.\,Pu(\phi u,\psi u) \overset{\pi_0}{\longmapsto} \lambda u.\,Pu\big(Pu(\phi u,\psi u),\,\star\big) \;=\; \lambda u.\,Pu(\phi u,\star) \;=\; E_0\phi. \qquad \blacksquare$$

Theorem 11.8 If $(\mathcal{C},\Sigma)$ is monadic and Σ has a constant then $\mathcal{C}$ is extensive, *i.e.* it has stable disjoint coproducts [Tay99, Section 5.5].

$$\begin{tikzcd}
X \arrow[r] \arrow[d] & U \arrow[d] & Y \arrow[l] \arrow[d] \\
\mathbf{1} \arrow[r] & \mathbf{2} & \mathbf{1} \arrow[l]
\end{tikzcd}$$

Proof. We have to show that the commutative diagram is a pair of pullbacks iff its top row is a coproduct. The map $U \to \mathbf{2}$ must arise from a prime $u : U \vdash Pu : \Sigma^{\Sigma\times\Sigma}$, from which Proposition 11.7 constructs subspaces forming pullbacks and a coproduct. Since pullbacks are unique up to isomorphism, if the given diagram is a pair of pullbacks then it is isomorphic to that in the Proposition.

Conversely, if we are given a coproduct $U = X + Y$ then $u : U \vdash Pu : \Sigma^{\Sigma\times\Sigma}$ is defined by $x : X \vdash Px \equiv \pi_0$ and $y : Y \vdash Py \equiv \pi_1$. Then E_0 in the Proposition becomes

$$E_0\theta \;=\; E_0\langle\phi,\psi\rangle \;=\; \big\langle \lambda x.\,\pi_0\langle\phi,\psi\rangle x,\ \lambda y.\,\star\big\rangle \;=\; \langle\phi,\star\rangle,$$

which agrees with Lemma 11.4. ■

Corollary 11.9 Let $x : X \vdash fx, gx : Z$ and $x : X \vdash Px : \Sigma^{\Sigma\times\Sigma}$ with P prime. Then

$$\mathsf{if}\,(\mathsf{focus}\,P)\,\mathsf{then}\,fx\,\mathsf{else}\,gx \;=\; \mathsf{focus}\,\big(\lambda\theta.\,Px(\theta(fx))(\theta(gx))\big). \qquad \blacksquare$$

Turning from coproducts to coequalisers, of course we can only construct those that are Σ-split. We leave the reader to formulate Beck-style equations as in Section 3, concentrating instead on the analogue for quotients of the comprehension calculus for subspaces in Section 8. Beware also that the class of Σ-epis is stable under products but not necessarily pullbacks. For the topological motivation, recall that the ***quotient topology*** on a set Y induced by the function $q : X \to Y$ from a topological space has $V \subset Y$ open iff $q^{-1}V$ is open in X.

Lemma 11.10 Let $X \underset{\widehat{Q}}{\overset{q}{\rightleftarrows}} Y$ be a Σ-split epi, and put $\widehat{R} = \widehat{Q}\cdot q$. Then

$$Y \;\cong\; \{\Sigma^2 X \mid E\},\quad \text{where } E\mathcal{F}F \;=\; F\big(\lambda x.\,\mathcal{F}(\lambda\phi.\,R\phi x)\big),$$

$$qx \;=\; \mathsf{admit}(\lambda\phi.\,R\phi x) \quad\text{and}\quad Q\phi \;=\; \lambda y.\,iy\phi.$$

Proof. Three of the four squares

$$\begin{tikzcd}[column sep=huge, row sep=huge]
\Sigma^2 X \arrow[r, two heads, shift left, "\Sigma^2 q"] \arrow[d, shift left, "\widehat{\eta}_{\Sigma^X}"] & \Sigma^2 Y \arrow[l, shift left, "\Sigma^Q"] \arrow[d, shift left, "\widehat{\eta}_{\Sigma^Y}"] \\
X \arrow[u, shift left, "\eta_X"] \arrow[r, two heads, shift left, "q"] & Y \arrow[l, shift left, "\widehat{Q}"] \arrow[u, shift left, "\eta_Y"]
\end{tikzcd}$$

commute by naturality, but that from Y to $\Sigma^2 X$ need not. So $\widehat{E}$ is the composite from $\Sigma^2 X$ anticlockwise back to itself:

$$\widehat{E} \;=\; \Sigma^2 q\,;\widehat{\eta}_{\Sigma^Y}\,;\eta_Y\,;\Sigma^Q \;=\; \widehat{\eta}_{\Sigma^X}\,;q\,;\eta_Y\,;\Sigma^Q \;=\; \widehat{\eta}_{\Sigma^X}\,;\eta_X\,;\Sigma^{\Sigma^q}\,;\Sigma^Q,$$

so $E\mathcal{F}F \;=\; \eta_{\Sigma^X}\big(\Sigma^{\eta_X}(\Sigma^2 R\mathcal{F})\big)F \;=\; F\big(\lambda x.\,\mathcal{F}(\lambda\phi.\,R\phi x)\big)$. Next,

$$i(qx) \;=\; \Sigma^Q\big(\eta_Y(qx)\big) \;=\; \Sigma^Q\big(\Sigma^2 q(\eta_X x)\big) \;=\; \Sigma^R(\eta_X) \;=\; \lambda\phi.\,R\phi x,$$

whence $qx = \mathsf{admit}(\lambda\phi.\,R\phi x)$. Finally, $iy\phi \;=\; \Sigma^Q(\eta_Y y)\phi \;=\; (\lambda\phi.\,\phi y)(Q\phi) \;=\; Q\phi y$. ■

Proposition 11.11 $Y \equiv X/R, \quad qx \equiv [x]$ and

$$\big(\mathsf{let}\,y=[x]\,\mathsf{in}\,fx\big) \;\equiv\; \mathsf{focus}\,\big(\lambda\theta.\,Q(\theta\cdot f)y\big) \;\equiv\; \mathsf{focus}\,\big(\lambda\theta.\,(iy)(\theta\cdot f)\big)$$

satisfy the rules

$$x:X \;\vdash\; [x]:X/R \qquad /I$$

$$\frac{\Gamma,\;x:X\;\vdash\;fx:Z \qquad\qquad \Gamma,\;\theta:\Sigma^Z\;\vdash\;(\theta\cdot f)=R(\theta\cdot f):\Sigma}{\Gamma,\;y:X/R\;\vdash\;\big(\mathsf{let}\,y=[x]\,\mathsf{in}\,fx\big):Z} \qquad /E$$

$$\frac{\Gamma,\;x:X\;\vdash\;fx:Z \qquad\qquad \Gamma,\;\theta:\Sigma^Z\;\vdash\;(\theta\cdot f)=R(\theta\cdot f):\Sigma}{\Gamma,\;x:X\;\vdash\;\big(\mathsf{let}\,[x]=[x']\,\mathsf{in}\,fx'\big)=fx} \qquad /\beta$$

$$y:X/R\;\vdash\;\big(\mathsf{let}\,y=[x]\,\mathsf{in}\,[x]\big)=y \qquad /\eta,$$

together with those for the quotient *topology* $(\Sigma^{/}I,\,\Sigma^{/}E,\,\Sigma^{/}\beta$ and $\Sigma^{/}\eta)$,

$$\begin{array}{rcl} \phi:\Sigma^X & \vdash & Q\phi:\Sigma^{X/R} \\ \psi:\Sigma^{X/R} & \vdash & \Sigma^q\psi:\Sigma^X \end{array} \qquad\qquad \begin{array}{rcl} \phi:\Sigma^X,\,x:X & \vdash & (Q\phi)(qx) \;=\; R\phi x \\ \psi:\Sigma^{X/R} & \vdash & Q(\Sigma^q\psi) \;=\; \psi. \end{array}$$

Proof. We verify the $/\beta$ and $/\eta$ rules:

$$\begin{array}{rcll}
\mathsf{let}\;[x]=[x']\,\mathsf{in}\,fx' & = & \mathsf{focus}\,\big(\lambda\theta.\,i(qx)(\lambda x'.\,\theta(fx'))\big) & \text{definition} \\
 & = & \mathsf{focus}\,\big(\lambda\theta.\,(\lambda\phi.\,R\phi x)(\lambda x'.\,\theta(fx'))\big) & \text{Lemma 11.10} \\
 & = & \mathsf{focus}\,\big(\lambda\theta.\,R(\theta\cdot f)x\big) & \\
 & = & \mathsf{focus}\,\big(\lambda\theta.\,(\theta\cdot f)x\big) \;=\; fx & R(\theta\cdot f)=(\theta\cdot f) \\
\mathsf{let}\,y=[x]\,\mathsf{in}\,[x] & = & \mathsf{focus}\,\big(\lambda\psi.\,(iy)(\psi\cdot q)\big) & \\
 & = & \mathsf{focus}\,\big(\lambda\psi.\,Q(\psi\cdot q)y\big) & \text{Lemma 11.10} \\
 & = & \mathsf{focus}\,\big(\lambda\psi.\,(Q\cdot\Sigma^q)\psi y\big) & \\
 & = & \mathsf{focus}(\lambda\psi.\,\psi y) \;=\; y & Q\cdot\Sigma^q=\mathsf{id}.\ ■
\end{array}$$

Remark 11.12 Notice that the β-rule is *not* a computational reduction, but a denotational equation that is a consequence of the equation $R(\theta\cdot f)=(\theta\cdot f)$ in its hypothesis. A *compiler* equipped with a proof assistant might perhaps be able know this, but otherwise we can only hope that the implementation will somehow find its way from one side of the β-rule to the other.

Let us specialise this to sets and (discrete) topological spaces, now making use of the lattice structure.

Example 11.13 Section C 10 constructs the quotient X/δ of an overt discrete object X by the open equivalence relation classified by $\delta : X \times X \to \Sigma$. Overt discrete means that X is equipped with predicates $\exists_X : \Sigma^X \to \Sigma$ and $(=_X) : X \times X \to \Sigma$ satisfying the usual properties.

The construction in Lemma C 10.8 obtains $\Sigma^{X/\delta}$ (which is unfortunately called $\Sigma^Q = E$ there, for "quotient" and "equaliser") as a retract of Σ^X, namely as the image of the closure operation

$$R \;\equiv\; \Sigma^q \cdot \exists_q \;\equiv\; \lambda\phi x.\, \exists y.\, \delta(x,y) \wedge \phi(y).$$

The quotient space is therefore $X/\delta = X/R = \{\Sigma^2 X \mid E\}$, where

$$\begin{array}{rcl} E\mathcal{F}F & \equiv & F\big(\lambda x.\, \mathcal{F}(\lambda\phi.\, \exists y.\, \delta(x,y) \wedge \phi y)\big) \\ q(x) & \equiv & \mathsf{admit}\,\big(\lambda\phi.\, \exists y.\, \delta(x,y) \wedge \phi(y)\big). \end{array}$$

If $f : X \to Z$ respects the equivalence relation δ then the β-rule is

$$\mathsf{focus}\,\big(\lambda\theta.\, \exists y.\, \delta(x,y) \wedge \theta(fy)\big) \;=\; fx.$$

In principle this computation involves a search of the equivalence class, which makes sense, given the connection between coequalisers and while programs [Tay99, Section 6.4].

Example 11.14 As explained in Remark C 10.12, there is another representation of X/δ that is more like the familiar one with equivalence classes. It arises from the factorisation

$$\begin{tikzcd} X \arrow[r, two heads, shift left, "q"] & X/\delta \arrow[l, tail, shift left, "\widehat{\exists}_q"] \arrow[r, shift left, "\{\}"] & \Sigma^{X/\delta} \arrow[l, tail, shift left] \arrow[r, shift left, "\Sigma^q"] \arrow[r, phantom, "\top"] & \Sigma^X \arrow[l, two heads, shift left, "\exists_q"] \end{tikzcd}$$

of the transpose $\tilde{\delta}$ of the equivalence relation. This generalises

$$\mathbb{N} \;\cong\; \{\Sigma^{\mathbb{N}} \mid E\} \text{ by } n \mapsto \{n\}, \quad \text{where} \quad E \;=\; \lambda F\phi.\, \exists n.\, F(\lambda m.\, m = n) \wedge \phi n,$$

cf. Section A 10. The quotient is now given by $\{\Sigma^X \mid E\}$, where

$$E \;=\; \lambda F\phi.\, \exists x.\, F\big(\lambda y.\, \delta(x,y)\big) \wedge \phi x,$$

with $qx = \mathsf{admit}\,\big(\lambda y.\, \delta(x,y)\big)$.

If $f : X \to Z$ respects the equivalence relation then the mediator $X/\delta \to Z$ is

$$y \;\mapsto\; \mathsf{focus}\,\big(\lambda\theta.\, \exists x'.\, iyx' \wedge \theta(fx')\big).$$

This selects an element x' from the equivalence class $y \in X/\delta$ (classified by a predicate $iy \in \Sigma^X$) and applies f to it, as we would expect. Substitution of $y = qx$ yields the same formula $\mathsf{focus}\,\big(\lambda\theta.\, \exists x'.\, \delta(x,x') \wedge \theta(fx')\big)$ as before. ∎

Remark 11.15 Here is a summary of the comprehension types that we have used.

$$\begin{array}{rcll} X & \cong & \{\Sigma^2 X \mid \lambda\mathcal{F}F.\, F\big(\lambda x.\, \mathcal{F}(\lambda\phi.\, \phi x)\big)\} & \text{Remark 4.11} \\ \{X \mid E_0\} \times \{Y \mid E_1\} & = & \{X \times Y \mid E_1^X \cdot E_0^Y\} & \text{Proposition 5.12} \\ \{X \mid E_0\} + \{Y \mid E_1\} & = & \{\Sigma^{\Sigma^X \times \Sigma^Y} \mid E\} & \text{Example 11.2} \\ \text{where} \;\; E\mathcal{H}H & = & H\big\langle \lambda x.\, \mathcal{H}(\lambda\phi\psi.\, E_0\phi x), \lambda y.\, \mathcal{H}(\lambda\phi\psi.\, E_1\phi y)\big\rangle & \\ \Sigma^{\{X \mid E\}} & = & \{\Sigma^X \mid \Sigma^E\} & \text{Proposition 6.2} \\ \{\{X \mid E_1\} \mid E_2\} & = & \{X \mid E_2\} & \text{Proposition 6.10} \\ \mathsf{pts}(A, \alpha) & = & \{\Sigma^A \mid \lambda F\phi.\, \phi(\alpha F)\} & \text{Example 11.1} \\ X/R & = & \{\Sigma^2 X \mid \lambda\mathcal{F}F.\, F\big(\lambda x.\, \mathcal{F}(\lambda\phi.\, R\phi x)\big)\} & \text{Proposition 11.11} \\ U \cap (X \setminus V) & = & \{X \mid \lambda\phi.\, U \wedge \phi \vee V\} & \text{Examples 2.5} \\ X/\delta & = & \{\Sigma^2 X \mid \lambda\mathcal{F}F.\, F\big(\lambda x.\, \mathcal{F}(\lambda\phi.\, \exists y.\, \delta(x,y) \wedge \phi y)\big)\} & \\ & \cong & \{\Sigma^X \mid \lambda F\phi.\, \exists x.\, F\big(\lambda y.\, \delta(x,y)\big) \wedge \phi x\} & \text{Examples 11.13f.} \end{array}$$

References

[Bou66] Nicolas Bourbaki. *Topologie Générale.* Hermann, 1966. Chapter I, "Structures Topologiques". English translation, "General Topology", distrubuted by Springer-Verlag, 1989.

[BR98] Anna Bucalo and Giuseppe Rosolini. Repleteness and the associated sheaf. *Journal of Pure and Applied Algebra*, 127:147–151, 1998.

[BW85] Michael Barr and Charles Wells. *Toposes, Triples, and Theories.* Number 278 in Grundlehren der mathematischen Wissenschaften. Springer-Verlag, 1985.

[Eck69] Beno Eckmann, editor. *Seminar on Triples and Categorical Homology Theory*, number 80 in Lecture Notes in Mathematics. Springer-Verlag, 1969.

[Fak70] Sabah Fakir. Monade idempotente associée à une monade. *Comptes Rendues de l'Academie des Sciences de Paris, Série A–B*, 270:A99–101, 1970.

[Fef77] Solomon Feferman. Categorical foundations and foundations of category theory. In Robert Butts and Jaakko Hintikka, editors, *Logic, Foundations of Mathematics and Computability Theory*, pages 149–169. Reidel, 1977.

[FS90] Peter Freyd and Andre Scedrov. *Categories, Allegories.* Number 39 in Mathematical Library. North-Holland, 1990.

[FW90] Barry Fawcett and Richard Wood. Completely distributive lattices I. *Mathematical Proceedings of the Cambridge Philosophical Society*, 107:81–9, 1990.

[GHK+80] Gerhard Gierz, Karl Heinrich Hoffmann, Klaus Keimel, Jimmie Lawson, Michael Mislove, and Dana Scott. *A Compendium of Continuous Lattices.* Springer-Verlag, 1980.

[HM81] Karl Hofmann and Michael Mislove. Local compactness and continuous lattices. In Bernhard Banaschewski and Rudolf-Eberhard Hoffmann, editors, *Continuous Lattices*, number 871 in Springer Lecture Notes in Mathematics, pages 209–248, 1981.

[Joh82] Peter Johnstone. *Stone Spaces.* Number 3 in Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1982.

[Law70] Bill Lawvere. Equality in hyperdoctrines and the comprehension schema as an adjoint functor. In Alex Heller, editor, *Applications of Categorical Algebra*, number 17 in Proceedings of Symposia in Pure Mathematics, pages 1–14. American Mathematical Society, 1970.

[LR73] Joachim Lambek and Basil Rattray. Localizations at injective objects in complete categories. *Proceedings of the American Mathematical Society*, 41:1–9, 1973.

[LR75] Joachim Lambek and Basil Rattray. Localizations and sheaf reflectors. *Transactions of the American Mathematical Society*, 210:279–293, 1975.

[ML71] Saunders Mac Lane. *Categories for the Working Mathematician.* Number 5 in Graduate Texts in Mathematics. Springer-Verlag, 1971.

[ML84] Per Martin-Löf. *Intuitionistic Type Theory.* Bibliopolis, Naples, 1984.

[Mog91] Eugenio Moggi. Notions of computation and monads. *Information and Computation*, 93:55–92, 1991.

[MRW01] Francisco Marmolejo, Robert Rosebrugh, and Richard Wood. A basic distributive law. *Journal of Pure and Applied Algebra*, 168:209–226, 2001.

[Par71] Robert Paré. On absolute colimits. *Journal of Algebra*, 19:80–95, 1971.

[Par74] Robert Paré. Colimits in topoi. *Bulletin of the American Mathematical Society*, 80(3):556–561, 1974.

[RR88] Edmund Robinson and Giuseppe Rosolini. Categories of partial maps. *Information and Computation*, 79:95–130, 1988.

[Sco72] Dana Scott. Continuous lattices. In Bill Lawvere, editor, *Toposes, Algebraic Geometry and Logic*, number 274 in Lecture Notes in Mathematics, pages 97–137. Springer–Verlag, 1972.

[Sco76] Dana Scott. Data types as lattices. *SIAM Journal on Computing*, 5:522–587, 1976.

[Tar35] Alfred Tarski. Zur Grundlegung der Boole'schen algebra. *Fundamenta Mathemematica*, 24:177–198, 1935.

[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. Also available as technical report ECS-LFCS-97-376.

[Vic95] Steven Vickers. Locales are not pointless. In Ian Mackie and Rajagopal Nagarajan, editors, *Theory and Formal Methods of Computing*, 1995.

[Vic01] Steven Vickers. The double powerlocale and exponentiation. Submitted to *Theory and Applications of Categories*, 2001.

The papers on abstract Stone duality may be obtained from

`www.cs.man.ac.uk/∼pt/ASD`

[A] Paul Taylor, Sober spaces and continuations. *Theory and Applications of Categories*, 10(12):248–299, 2002.

[C] Paul Taylor, Geometric and higher order logic using abstract Stone duality. *Theory and Applications of Categories*, 7(15):284–338, 2000.

[D] Paul Taylor, Non-Artin gluing in recursion theory and lifting in abstract Stone duality. 2000.

[E–] Paul Taylor, Local compactness and the Baire category theorem in abstract Stone duality. *Category Theory and Computer Science*, Ottawa, 2002. *ENTCS* 69.

[E] Paul Taylor, Computably based locally compact spaces. May 2003.

[F] Paul Taylor, Scott domains in abstract Stone duality. March 2002.

[G] Paul Taylor, An elementary theory of the category of locally compact locales. March 2003.