Proof Theory of Higher-Order Equations: 
Conservativity, Normal Forms 
and Term Rewriting

K. Meinke. 
Department of Numerical Analysis and Computer Science, 
KTH Stockholm, 
100 44 Stockholm, Sweden. 
October 25, 1999

Abstract

We introduce a necessary and sufficient condition for the \(\omega\)-extensionality rule of higher-order equational logic to be conservative over first-order many-sorted equational logic for ground first-order equations. This gives a precise condition under which computation in the higher-order initial model by term rewriting is possible. The condition is then generalised to characterise a normal form for higher-order equational proofs in which extensionality inferences occur only as the final proof inferences.

The main result is based on a notion of observational equivalence between higher-order elements induced by a topology of finite information on such elements. Applied to extensional higher-order algebras with countable first-order carrier sets, the finite information topology is metric and second countable in every type.

1 Introduction.

Higher-order equations have applications in diverse areas of computer science, such as specification languages, logics of programs, and declarative programming languages. The finite type system of Church [1940], which allows the construction of function types \((\sigma \rightarrow \tau)\) and product types \((\sigma \times \tau)\), forms the kernel of most higher-order type systems currently found in the literature. The model theory of higher-order equations and the theory of higher-order algebra, based on finite types, were considered in Meinke [1992].

Calculi for higher-order equations extend the first-order many-sorted equational calculus with additional extensionality rules, associated with the function space and product space type constructors. Such rules can either be finitary or infinitary in form, and have different completeness properties with respect to classes of higher-order algebras.

In Meinke [1992] it is shown that every set \(E\) of higher-order equations admits an extensional model \(I\) which is initial in the class of all minimal extensional models of \(E\). This model is termed the higher-order initial model of \(E\). Thus one can use higher-order equations as an algebraic specification language in an entirely analogous way to first-order equations. (See for example Ehrig and Mahr [1985].) The expressive power of higher-order equations under higher-order initial semantics, measured recursion theoretically, is much greater than the power of first-order equations under both first-order initial and first-order final semantics. (See Kosiczenko and Meinke [1995] and Meinke [1996].) So a definite technical advantage can be gained from the
higher-order approach. Now considering the higher-order initial model $I$ of a set $E$ of higher-order equations, one may pose the following

**Computation Problem:** *when can we compute in $I$ using the equations of $E$ oriented as rewrite rules?*

More precisely, given an equation $t = t'$, for ground (i.e. variable free) first-order terms $t$ and $t'$, which is true in $I$, when can we formally prove $t = t'$ by equational reasoning (and therefore term rewriting) alone, i.e. without using any of the extensionality rules?

This computation problem can be rephrased as a special case of more general proof theoretic questions about the conservativity properties of the extensionality rules over the underlying first-order many-sorted equational calculus. For both the finitary and infinitary extensionality rules, two conservativity properties are of obvious interest, namely conservativity over equational logic for:

(i) ground first-order equations, and

(ii) all first-order equations (i.e. allowing variables in terms).

The latter conservativity property is very strong, and leads to a normal form for higher-order equational proofs (which we term *eval normal form*) in which each extensionality rule is used at most once, and among the final proof inferences only. Since the higher-order initial model is constructed as a quotient term model using the infinitary $\omega$-extensionality rule, our computation problem above is equivalent to the following proof theoretic

**Conservativity Problem:** *when is the infinitary $\omega$-extensionality rule conservative over many-sorted first-order equational logic for ground first-order equations?*

The various conservativity properties of the extensionality rules of higher-order equational logic have considerable computational significance. On the one hand, these rules seem to lack any efficient implementation in a computational logic. On the other hand, when the extensionality rules are conservative over equational logic for some class of equations, then such equations can be derived (at least in principle) using the relatively efficient and well understood computational methods of term rewriting.

The main theoretical problem turns out to be characterising the conservativity of the infinitary $\omega$-extensionality rule. Not only does conservativity of this rule relate the higher-order initial model with term rewriting computation, there is also a direct relationship between conservativity of the infinitary and finitary extensionality rules. Therefore we can apply our main result characterising conservativity for the infinitary rule to characterise conservativity of the finitary extensionality rule as well, and at the same time characterise the existence of eval normal form proofs.

We will show that the conservativity of the infinitary extensionality rule depends upon finitistic properties of the higher-order operators themselves, in particular the property that higher-order operators use just a finite amount of information about their higher-order arguments to determine their values. More precisely, we formulate a notion of *observational equivalence* for elements of higher-order type based on a topology of *finite information* for higher-order types. This topology appears to be new in the literature. It can be applied to both extensional and non-extensional models of a higher-order signature $\Sigma$. Two elements of a $\Sigma$ algebra $A$ are observationally equivalent if they belong to precisely the same open sets in this topology. Our main result is:

**4.8. Conservativity Theorem.** Let $\Sigma$ be a higher-order signature which contains the homeomorphism operators and let $E$ be a higher-order equational theory which contains the homeomorphism axioms $\text{Hom}$. Then infinitary higher-order equational logic is conservative over equational
logic on $E$ for ground first-order equations if, and only if, observational equivalence $\equiv^{obs}$ is a congruence on the (first-order) initial model $I(\Sigma, E)$.

and this theorem is applied to the finitary extensionality rule to yield the following:

4.12. Normal Form Theorem. Let $\Sigma$ be a higher-order signature which contains the homeomorphism operators and let $E$ be a higher-order equational theory which contains the homeomorphism axioms $\text{Hom}$. The following are equivalent:
(i) finitary higher-order equational logic is conservative on $E$ for first-order equations;
(ii) for every equation $e \in \text{Eqn}(\Sigma, X)$ (of any order), if $Ee$ then there is a proof of $e$ which is in eval normal form;
(iii) observational equivalence, $\equiv^{obs}$, is a congruence on the free algebra $T_E(\Sigma, X)$.

In fact, continuity of the initial algebra $I(\Sigma, E)$ and the free algebra $T_E(\Sigma, X)$ with respect to the finite information topology (i.e. continuity of all operations of $I(\Sigma, E)$ and $T_E(\Sigma, X)$) are sufficient (but not necessary) conditions to ensure that observational equivalence is a congruence in both cases. At first sight, it may appear odd that the first-order initial model plays a role here, but recall that this is the unique minimal model where a ground equation is true precisely when it is formally provable in the first-order many-sorted equational calculus.

The topology of finite information is well known for first and second-order types. For first-order types it is (trivially) the discrete topology, while for second-order types it is the product or Tychonoff topology on the function space given the discrete topology on the domain and codomain sets. However, the product topology construction cannot be iterated for higher-order types in such a way that both function currying and function evaluation are continuous. The finite information topology introduced here has strong separation and countability properties. On extensional models with countable first-order carrier sets, it is metric and second countable in every type. The uniformity of these properties in all types imply that the finite information topology is not homeomorphic, for every type, with the well known Kleene-Kreisel topology on total functionals of finite type (see for example Normann [1980]). For example, the Kleene-Kreisel topology is not second countable above second-order types. In the finite information topology, evaluation and currying of functionals are continuous in all types. The topology is constructed using particular types which we term elementary or hereditarily Horn (following the types as propositions analogy). The elementary types are inductively defined types of the form $\beta, (\sigma \times \tau)$, and $(\sigma \to \beta)$ where $\beta$ is a basic (atomic) type and $\sigma$ and $\tau$ are elementary or hereditarily Horn types. In an extensional model, every space is homeomorphic to a space of elementary type.

The structure of this paper is as follows. In Section 2, we review the finitary and infinitary proof systems for higher-order equations and make precise the conservativity properties of interest. We consider how conservativity leads to the eval normal form for higher-order equational proofs in which extensionality inferences are the final inferences only. In Section 3, we introduce the finite information topology and establish its separation and countability properties. We prove the continuity of evaluation and currying in this topology. In Section 4, we characterise necessary and sufficient conditions for the extensionality rules to be conservative over equational logic for both ground and arbitrary first-order equations by means of the finite information topology. Several of the more lengthy but tedious proofs of Section 3 are consigned to the Appendix.

The main prerequisites of this paper are a familiarity with first-order many-sorted equational logic (see for example Taylor [1979] or Meinke and Tucker [1993]), term rewriting (see for example Klop [1993] or Dershowitz and Jouanneaud [1990]) and point set topology (see for example Dugundji [1966] or Kelley [1955]). While some familiarity with higher-order universal algebra is useful (a suitable introduction is Meinke [1992]), the paper is largely self contained on this subject.
2 Higher–Order Equational Logic.

We review some of the fundamental definitions and results of higher-order equational logic, including the proof systems introduced in Meinke [1992].

By a set $S$ of sorts we mean any non-empty set. As usual, $S^*$ denotes the set of all words in the free monoid generated by $S$. The empty word is denoted by $\lambda$ and $S^+ = S^* - \{ \lambda \}$ denotes the set of all non-empty words over $S$. An $S$-sorted signature $\Sigma$ is an $S^* \times S$ indexed family of disjoint sets $\Sigma = \langle \Sigma_{w,s} \mid w \in S^*, s \in S \rangle$. For the empty word $\lambda$ and each sort $s \in S$, each element $c \in \Sigma_{\lambda,s}$ is a constant symbol of sort $s$. For each non-empty word $w = s(1) \ldots s(n) \in S^+$ and each sort $s \in S$, each element $f \in \Sigma_{w,s}$ is a function symbol of domain type $w$, codomain type $s$ and arity $n$. Let $S$ be any sort set, $\Sigma$ be any $S$-sorted signature, and let $X = \langle X_s \mid s \in S \rangle$ be any $S$-indexed family of sets of variable symbols. (We normally assume that the sets $\Sigma_{\lambda,s}$ and $X_s$ are disjoint for each $s \in S$.) We let $T(\Sigma, X)_s$ denote the set of all terms over $\Sigma$ and $X$ of sort $s \in S$.

Let $\Sigma$ be an $S$-sorted signature. An $S$-sorted $\Sigma$ algebra is a pair $(A, \Sigma^A)$, consisting of an $S$-indexed family $A = \langle A_s \mid s \in S \rangle$ of sets termed the carrier sets of $A$, and an $S^* \times S$ indexed family

$$\Sigma^A = \langle \Sigma^A_{w,s} \mid w \in S^*, s \in S \rangle$$

of sets of constants and algebraic operations. For each sort $s \in S$, $\Sigma^A_{\lambda,s} = \langle c_A \mid c \in \Sigma_{\lambda,s} \rangle$, where $c_A \in A_s$ is a constant that interprets $c$ in $A_s$. For each word $w = s(1) \ldots s(n) \in S^+$ and each sort $s \in S$, $\Sigma^A_{w,s} = \langle f_A \mid f \in \Sigma_{w,s} \rangle$, where $f_A : A^w \to A_s$ is a function with domain $A^w = A_{s(1)} \times \ldots \times A_{s(n)}$ and codomain $A_s$ which interprets $f$ over $A$. As usual, we let $A$ denote both a $\Sigma$ algebra and its $S$-indexed family of carrier sets. We let $\text{Alg}(\Sigma)$ denote the class of all $S$-sorted $\Sigma$ algebras. We let $T(\Sigma, X)$ denote the free term algebra on the family $X$ of sets of generators, and $T(\Sigma) = T(\Sigma, \emptyset)$ denotes the absolutely free or ground term algebra on the $S$-indexed family $\emptyset$ of empty sets of generators. Recall that $T(\Sigma)$ is initial in $\text{Alg}(\Sigma)$, i.e. there exists a unique homomorphism from $T(\Sigma)$ to each algebra $A \in \text{Alg}(\Sigma)$. A $\Sigma$ algebra $A$ is minimal if, and only if, $A$ has no proper subalgebra, for example $T(\Sigma)$ is minimal.

Higher-order signatures and algebras are defined over the following system of types, often known as the system of finite or simple types (Church [1940]).

2.1. Definition. By a type basis $B$ we mean any non-empty set. The (finite) type hierarchy $H(B)$ generated by a type basis $B$ is the set $H(B) = \bigcup_{n \in \omega} H_n(B)$ defined inductively by

$$H_0(B) = B,$$

and

$$H_{n+1}(B) = H_n(B) \cup \{ (\sigma \times \tau), (\sigma \to \tau) \mid \sigma, \tau \in H_n(B) \}.$$

Each element $\sigma \in B$ is termed a basic type; each element $(\sigma \times \tau) \in H(B)$ is termed a product type and each element $(\sigma \to \tau) \in H(B)$ is termed a function type.

We can assign an order to each type $\sigma \in H(B)$ as follows. Each basic type $\sigma \in B$ has order 0. If $\sigma, \tau \in H(B)$ have order $m$ and $n$ respectively then $(\sigma \times \tau)$ has order $\text{sup}\{ m, n \}$ and $(\sigma \to \tau)$ has order $\text{sup}\{ m + 1, n \}$. A type structure $S$ over a type basis $B$ is a subset $S \subseteq H(B)$, which is closed under subtypes in the sense that for any $\sigma, \tau \in H(B)$, if $(\sigma \to \tau) \in S$ or $(\sigma \times \tau) \in S$ then both $\sigma \in S$ and $\tau \in S$.

A higher-order signature is simply an $S$-sorted signature $\Sigma$ in which $S$ is a type structure and $\Sigma$ contains distinguished operation symbols for the product and function types of $S$ as follows.
2.2. Definition. Let $S$ be a type structure over a type basis $B$. An $S$–typed signature $\Sigma$ is an $S$–sorted signature such that for each product type $\langle \sigma \times \tau \rangle \in S$ we have left and right projection operation symbols

$$
\text{proj}^{\langle \sigma \times \tau \rangle, \sigma} \in \Sigma_{\langle \sigma \times \tau \rangle, \sigma}, \quad \text{proj}^{\langle \sigma \times \tau \rangle, \tau} \in \Sigma_{\langle \sigma \times \tau \rangle, \tau}.
$$

also for each function type $\langle \sigma \to \tau \rangle \in S$ we have a binary evaluation operation symbol

$$
\text{eval}^{\langle \sigma \to \tau \rangle} \in \Sigma_{\langle \sigma \to \tau \rangle, \sigma, \tau}.
$$

When the types $\sigma$ and $\tau$ are clear, we let $\text{proj}^1$ and $\text{proj}^2$ denote $\text{proj}^{\langle \sigma \times \tau \rangle, \sigma}$ and $\text{proj}^{\langle \sigma \times \tau \rangle, \tau}$, and we let $\text{eval}$ denote $\text{eval}^{\langle \sigma \to \tau \rangle}$. Furthermore, we will often write terms of the form $\text{eval}(t, t')$ using the meta-notation $t(t')$ (applicative form) thereby omitting the evaluation operation symbol which can be inferred from the types of $t$ and $t'$.

We can now introduce the intended interpretations of an $S$–typed signature $\Sigma$.

2.3. Definition. Let $S$ be a type structure over a type basis $B$. Let $\Sigma$ be an $S$–typed signature and $A$ be an $S$–sorted $\Sigma$ algebra. We say that $A$ is an $S$–typed $\Sigma$ algebra if, and only if, for each product type $\langle \sigma \times \tau \rangle \in S$ we have $A_{\langle \sigma \times \tau \rangle} \subseteq A_{\sigma} \times A_{\tau}$, and the mappings

$$
\text{proj}^{\langle \sigma \times \tau \rangle, \sigma}_{\Sigma} : A_{\langle \sigma \times \tau \rangle} \to A_{\sigma}, \quad \text{proj}^{\langle \sigma \times \tau \rangle, \tau}_{\Sigma} : A_{\langle \sigma \times \tau \rangle} \to A_{\tau}
$$

are the left and right projection mappings defined on $A_{\langle \sigma \times \tau \rangle}$ by

$$
\text{proj}^{\langle \sigma \times \tau \rangle, \sigma}_{\Sigma}((a_1, a_2)) = a_1, \quad \text{proj}^{\langle \sigma \times \tau \rangle, \tau}_{\Sigma}((a_1, a_2)) = a_2,
$$

for any pair $(a_1, a_2) \in A_{\langle \sigma \times \tau \rangle}$. Furthermore, for each function type $\langle \sigma \to \tau \rangle \in S$ we have $A_{\langle \sigma \to \tau \rangle} \subseteq [A_{\sigma} \to A_{\tau}]$, and the operation $\text{eval}^{\langle \sigma \to \tau \rangle}_{\Sigma} : A_{\langle \sigma \to \tau \rangle} \times A_{\sigma} \to A_{\tau}$ is the evaluation mapping on the function space $A_{\langle \sigma \to \tau \rangle}$ defined by

$$
\text{eval}^{\langle \sigma \to \tau \rangle}_{\Sigma}(a, b) = a(b),
$$

for each $a \in A_{\langle \sigma \to \tau \rangle}$ and $b \in A_{\sigma}$. □

In the remainder of this section, unless stated otherwise, we let $S$ denote a fixed, but arbitrarily chosen type structure over a type basis $B$ and we let $\Sigma$ denote a fixed, arbitrarily chosen $S$-typed signature. We let $X = \{X_\tau \mid \tau \in S\}$ denote an $S$–indexed family of disjoint, infinite sets $X_\tau$ of variable symbols of type $\tau$.

Within the class $\text{Alg}(\Sigma)$ of all algebras of signature $\Sigma$, it is important to distinguish between those algebras which are extensional, and those which are non-extensional. We say that a $\Sigma$ algebra $A$ is extensional if, and only if, $A$ satisfies the $\Sigma$ sentences:

$$
\forall x \forall y \left( \text{proj}^1(x) = \text{proj}^1(y) \land \text{proj}^2(x) = \text{proj}^2(y) \Rightarrow x = y \right) \label{extensional1}
$$

for each product type $\langle \sigma \times \tau \rangle \in S$, and

$$
\forall x \forall y \left( \forall z \left( \text{eval}(x, z) = \text{eval}(y, z) \Rightarrow x = y \right) \right) \label{extensional2}
$$

for each function type $\langle \sigma \to \tau \rangle \in S$. We let $\text{Ext}$ denote the set of all extensionality sentences of the forms (1) and (2) above, and we let $\text{Alg}_{\text{Ext}}(\Sigma)$ denote the class of all extensional $\Sigma$ algebras.
Clearly, every \(S\)-typed \(\Sigma\) algebra is extensional. The significance of the distinction between extensional and non-extensional algebras can be summarised by the following

2.4. Collapsing Theorem. (Shepherdson, Mostowski) Let \(A\) be an \(S\)-sorted \(\Sigma\) algebra. Then \(A\) is isomorphic to an \(S\)-typed \(\Sigma\) algebra if, and only if, \(A\) is extensional.

Proof. See Meinke [1992].

Thus we study the intended models of \(\Sigma\), up to isomorphism, as the extensional models of \(\Sigma\).

By a higher-order equation over \(\Sigma\) and \(X\) of type \(\tau \in S\) we mean a formula of the form \(t = t'\) where \(t, t' \in T(\Sigma, X)_{\tau}\) are terms of type \(\tau\). If \(\tau\) is an \(n\)-th order type then we may say that \(t = t'\) is an \(n\)-th order equation. We let \(\text{Eqn}(\Sigma, X)_{\tau}\) denote the set of all equations over \(\Sigma\) and \(X\) of type \(\tau\) and \(\text{Eqn}(\Sigma, X) = \bigcup_{\tau \in S} \text{Eqn}(\Sigma, X)_{\tau}\). An equation \(t = t' \in \text{Eqn}(\Sigma, X)_{\tau}\) is said to be ground if, and only if, the terms \(t\) and \(t'\) are ground terms, i.e. \(t, t' \in T(\Sigma)_{\tau}\).

We let \(\text{Alg}(\Sigma, E)\) denote the class of all \(\Sigma\) algebras which are models of \(E\),

\[
\text{Alg}(\Sigma, E) = \{ A \in \text{Alg}(\Sigma) : A \models E \},
\]

and we let \(\text{Alg}_{\text{Ext}}(\Sigma, E)\) denote the class of all extensional \(\Sigma\) algebras which are models of \(E\),

\[
\text{Alg}_{\text{Ext}}(\Sigma, E) = \{ A \in \text{Alg}_{\text{Ext}}(\Sigma) : A \models E \}.
\]

We can construct a sound and complete calculus for any higher-order equational theory \(E\) with respect to the class \(\text{Alg}_{\text{Ext}}(\Sigma, E)\) of all extensional models of \(E\) using just finitary deduction rules. For this we add to the deduction rules of many-sorted equational logic additional rules which incorporate the extensionality axiom schemas (1) and (2) above, as follows.

2.5. Definition. The finitary deduction rules of higher-order equational logic are the following.

(i) For any type \(\tau \in S\) and any term \(t \in T(\Sigma, X)_{\tau}\),

\[
\frac{t = t}{t = t}
\]

is a reflexivity rule.

(ii) For any type \(\tau \in S\) and any terms \(t_0, t_1 \in T(\Sigma, X)_{\tau}\),

\[
\frac{t_0 = t_1}{t_1 = t_0}
\]

is a symmetry rule.

(iii) For any type \(\tau \in S\) and any terms \(t_0, t_1, t_2 \in T(\Sigma, X)_{\tau}\),

\[
\frac{t_0 = t_1, \ t_1 = t_2}{t_0 = t_2}
\]

is a transitivity rule.

(iv) For each type \(\tau \in S\), any terms \(t, t' \in T(\Sigma, X)_{\sigma}\), any type \(\sigma \in S\), any variable symbol \(x \in X_{\sigma}\) and any terms \(t_0, t_1 \in T(\Sigma, X)_{\sigma}\),

\[
\frac{t = t', \ t_0 = t_1}{t[x/t_0] = t'[x/t_1]}
\]

is a substitution rule. (As usual, \(t_i[x/t_j]\) denotes the result of substituting the variable \(x\) by the term \(t_j\) uniformly in \(t_i\) when \(x\) and \(t_j\) have the same sort.)
(v) For each product type \((\sigma \times \tau) \in S\) and any terms \(t_0, t_1 \in T(\Sigma, X)_{(\sigma \times \tau)}\),

\[
\begin{align*}
proj^1(t_0) &= proj^1(t_1), \\
proj^2(t_0) &= proj^2(t_1)
\end{align*}
\]

\(t_0 = t_1\)

is a projection rule.

(vi) For each function type \((\sigma \to \tau) \in S\), any terms \(t_0, t_1 \in T(\Sigma, X)_{(\sigma \to \tau)}\) and any variable symbol \(x \in X_\sigma\) not occurring in \(t_0\) or \(t_1\),

\[
\begin{align*}
eval(t_0, x) &= eval(t_1, x)
\end{align*}
\]

\(t_0 = t_1\)

is a (finitary) extensionality rule. \(\square\)

Note that in each of the above deduction rules both the conclusion and each of the premises is an equation. In particular, the calculus is quantifier free.

By a proof of an equation \(e \in Eqn(\Sigma, X)\) from a set \(E \subseteq Eqn(\Sigma, X)\) of equations using the finitary deduction rules of higher-order equational logic, we mean a finitely branching rooted tree \(P\) of finite depth with each node \(n\) labelled by an equation \(e_n \in Eqn(\Sigma, X)\) such that the root of \(P\) is labelled by \(e\), and for each node \(n\) in \(P\), either \(n\) has no antecedent nodes and \(e_n \in E\) is an axiom, or \(n\) has exactly \(k\) antecedent nodes \(m_1,\ldots,m_k\) for \(0 \leq k \leq 2\) and

\[
eq\frac{e_{m_1},\ldots,e_{m_k}}{e_n}
\]

is a finitary deduction rule of higher-order equational logic. (In the sequel, we also consider infinitary proofs using infinitary deduction rules.)

The finitary deduction rules of higher-order equational logic induce an inference relation, denoted by \(\vdash_{eval}\), between equational theories \(E \subseteq Eqn(\Sigma, X)\) and equations \(e \in Eqn(\Sigma, X)\), defined by \(E \vdash_{eval} e\) if, and only if, there exists a proof of \(e\) from \(E\) using the finitary deduction rules of higher-order equational logic alone. We shall reserve the symbol \(\vdash\) to denote the inference relation induced by the rules of first-order many-sorted equational logic. Thus \(E \vdash e\) if, and only if, there exists a proof of \(e\) from \(E\) (in the above sense) using only rules 2.5.(i)-(iv).

The completeness theorem for first-order single-sorted equational logic is due to Birkhoff [1935]. The problems of the many-sorted case, i.e. the possibility of an empty carrier set for some sort, and the consequent unsoundness of rules 2.5.(i)-(iv) above, were considered in Goguen and Meseguer [1982]. We can avoid the soundness problems of the many-sorted case here by the simplifying assumption on \(\Sigma\) that we can form a ground \(\Sigma\)-term of each type \(\tau \in S\). In this case we say that \(\Sigma\) is non-void. The completeness theorem for higher-order equational logic is then similar in style and proof to the first-order case.

**2.6. Completeness Theorem.** Let \(E \subseteq Eqn(\Sigma, X)\) be any higher-order equational theory. If \(\Sigma\) is non-void then for any higher-order equation \(e \in Eqn(\Sigma, X)\),

\[
E \vdash_{eval} e \iff Alg_{Ext}(\Sigma, E) \models e.
\]

**Proof.** See Meinke [1992]. \(\square\)

Recall that for any equational theory \(E\), the class \(Alg(\Sigma, E)\) contains a free algebra \(T_E(\Sigma, X)\) generated by an \(S\)-indexed family \(X\) of sets of generators. In particular, \(Alg(\Sigma, E)\) contains an initial or absolutely freely generated algebra \(I(\Sigma, E) = T_E(\Sigma, \emptyset)\). Now in general, for a higher-order equational theory \(E\), the class \(Alg_{Ext}(\Sigma, E)\) of all extensional models will not contain an
initial algebra. (Consider that this class need not be closed under subalgebras.) However, it is obvious that $\text{Alg}_{\text{Ext}}(\Sigma, E)$ contains at least one minimal algebra, namely the unit $\Sigma$ algebra. Furthermore, using purely model theoretic constructions, it is possible to show that the non-empty class $\text{Min}_{\text{Ext}}(\Sigma, E) \subseteq \text{Alg}_{\text{Ext}}(\Sigma, E)$ of all minimal extensional models of $E$ contains an initial algebra $I_{\text{Ext}}(\Sigma, E)$. By initiality, there exists a unique homomorphism from $I_{\text{Ext}}(\Sigma, E)$ to each algebra $A \in \text{Min}_{\text{Ext}}(\Sigma, E)$, and $I_{\text{Ext}}(\Sigma, E)$ is unique up to isomorphism. This algebra is termed the higher-order initial model of $E$. For computer science, it provides a suitable minimal model semantics for $E$ viewed as an equational specification of some computational system. Examples of the use of higher-order equations as system specifications may be found in Meinke [1994], Meinke and Steggles [1994] and Steggles [1996].

The higher-order initial model $I_{\text{Ext}}(\Sigma, E)$ cannot usually be constructed as the quotient of the term algebra $T(\Sigma)$ factored by the deductive closure of $E$ using the finitary calculus of Definition 2.5, the recursion theoretic complexity of the model may be too great for this construction to be possible. (See Kosiuczenko and Meinke [1995] and Meinke [1996], which compare the complexity of first and higher-order initial models.) Instead, the following infinitary version of the extensionality rule 2.5.(vi) is required to make a quotient term model construction of $I_{\text{Ext}}(\Sigma, E)$.

2.7. Definition. For each function type $(\sigma \to \tau) \in S$ and any terms $t_0, t_1 \in T(\Sigma, X)_{(\sigma \to \tau)}$,

$$\langle \text{eval}(t_0), t \rangle = \text{eval}(t_1), t \mid t \in T(\Sigma)_\sigma$$

is an (infinitary) $\omega$–extensionality rule. If $\tau \in B$ is a basic type then this rule is also termed a basic $\omega$–extensionality rule. □

By a proof of an equation $e \in \text{Eqn}(\Sigma, X)$ from a set $E \subseteq \text{Eqn}(\Sigma, X)$ of equations using the infinitary rules of higher-order equational logic we mean the obvious generalisation of a finitary proof $P$ allowing the use of rule schemes 2.5.(i)-(v) together with infinite branching and the use of the $\omega$–extensionality rule for each function type $(\sigma \to \tau) \in S$ instead of rule scheme 2.5.(vi). For use in Section 4, we define here the degree $\text{deg}(P) \in \text{Ord}$ of an infinitary higher-order equational proof $P$ to be the ordinal depth of nesting of $\omega$-extensionality inferences in $P$. Thus $\text{deg}(P) = 0$ for any proof $P$ consisting of a single equational axiom $e \in E$. If the final inference in $P$ uses one of the rule schemes 2.5.(i)-(v) applied to subproofs $P_1, \ldots, P_i$ (for $i = 0, 1, 2$) then $\text{deg}(P) = \sup \{ \text{deg}(P_1), \ldots, \text{deg}(P_i) \}$. If the final inference in $P$ uses an $\omega$-extensionality rule applied to a family $\langle P(t_0) \mid t_0 \in T(\Sigma)_\tau \rangle$ of subproofs, then

$$\text{deg}(P) = 1 + \sup \{ \text{deg}(P(t_0)) \mid t_0 \in T(\Sigma)_\tau \}.$$ 

We define the inference relation $\vdash_\omega$ between higher-order equational theories $E \subseteq \text{Eqn}(\Sigma, X)$ and higher-order equations $e \in \text{Eqn}(\Sigma, X)$ by $E \vdash_\omega e$ if, and only if, there exists a proof of $e$ from $E$ using the infinitary rules of higher-order equational logic. If $E \vdash_\omega e$ and there exists a proof $P$ of $e$ of degree $\alpha \in \text{Ord}$, then we may write $E \vdash_{\omega, \alpha} e$.

Define the $S$–indexed family $\equiv^{E, \omega}_\tau = \langle \equiv^{E, \omega}_\tau \mid \tau \in S \rangle$ of binary relations $\equiv^{E, \omega}_\tau$ on terms in $T(\Sigma)_\tau$ by

$$t \equiv^{E, \omega}_\tau t' \iff E \vdash_\omega t = t',$$

for each type $\tau \in S$ and any terms $t, t' \in T(\Sigma)_\tau$. Clearly, rules 2.5.(i)-(iv) ensure that $\equiv^{E, \omega}_\tau$ is a congruence on the ground term algebra $T(\Sigma)$. Then $\equiv^{E, \omega}_\tau$ gives the following quotient term model construction.

2.8. Lemma. Let $E \subseteq \text{Eqn}(\Sigma, X)$ be any higher-order equational theory. The quotient term algebra $T(\Sigma)/\equiv^{E, \omega}_\tau$ is initial in $\text{Min}_{\text{Ext}}(\Sigma, E)$. 

8
Proof. See Meinke [1992].

In the sequel, we let $I_{\text{Ext}}(\Sigma, E)$ denote the quotient term algebra $T(\Sigma)/\equiv^{E,\omega}$. The higher-order initial model can be used to obtain the following completeness result for infinitary higher-order equational logic.

2.9. Completeness Theorem. Let $E \subseteq \text{Eqn}(\Sigma, X)$ be any higher-order equational theory. If $\Sigma$ is non-void then for any ground equation $e \in \text{Eqn}(\Sigma, X)$,

$$E \vdash_{\omega} e \iff \text{Min}_{\text{Ext}}(\Sigma, E) \models e.$$  

Proof. $\Rightarrow$ By induction on the complexity of infinitary proofs. $\Leftarrow$ Follows from the fact that $I_{\text{Ext}}(\Sigma, E) \in \text{Min}_{\text{Ext}}(\Sigma, E)$ and $I_{\text{Ext}}(\Sigma, E)$ is generic for ground equations, i.e., for any ground equation $e \in \text{Eqn}(\Sigma, X)$,

$$E \vdash_{\omega} e \iff I_{\text{Ext}}(\Sigma, E) \models e.$$  

Note that Theorem 2.9 does not hold if variables are allowed in equations. To obtain completeness in this case the full $\omega$-rule of equational logic is required. (See for example Meinke and Tucker [1992].)

Having presented the finitary and infinitary calculi for higher-order equations, and their completeness properties, we can now make precise the conservativity properties of interest in this paper.

2.10. Definition. Let $E \subseteq \text{Eqn}(\Sigma, X)$ be a higher-order equational theory and let $K \subseteq \text{Eqn}(\Sigma, X)$ be any class of equations.

(i) We say that finitary higher-order equational logic is conservative (over equational logic) on $E$ for $K$ equations if, and only if, for every equation $e \in K$,

$$E \vdash_{\text{eval}} e \Rightarrow E \vdash e.$$  

(ii) We say that infinitary higher-order equational logic is conservative (over equational logic) on $E$ for $K$ equations if, and only if, for every equation $e \in K$,

$$E \vdash_{\omega} e \Rightarrow E \vdash e.$$  

We are primarily interested in conservativity in the cases where $K$ is:

(i) the class of all ground first-order $\Sigma$ equations, and

(ii) the class of all first-order $\Sigma$ equations.

Obviously, if $E \vdash_{\text{eval}} e$ then $E \vdash_{\omega} e$, although the converse need not hold. Thus if infinitary higher-order equational logic is conservative on $E$ for $K$ equations then finitary higher-order equational logic is also conservative on $E$ for $K$ equations, but the converse need not hold. We are primarily interested in the conservativity of infinitary higher-order equational logic for classes of equations. This property is intimately connected with term rewriting computation and equational theorem proving in the higher-order initial model by virtue of the following proposition.

2.11. Proposition. Let $E \subseteq \text{Eqn}(\Sigma, X)$ be any higher-order equational theory and suppose that $\Sigma$ is non-void. Then the following are equivalent:

(i) infinitary higher-order equational logic is conservative on $E$ for ground first-order equations,
(ii) for any ground first-order equation $e \in \text{Eqn}(\Sigma, X)$,

$$I_{\text{Ext}}(\Sigma, E) \models e \iff E \vdash e.$$  

**Proof.** Immediate from Lemma 2.8 and Definition 2.10. \hfill \square

Let us consider examples of higher-order equational theories for which infinitary higher-order equational logic is conservative over equational logic for various types of equations.

**2.12. Examples.** (i.a). Let $\Sigma$ be any $S$-typed signature over a type basis $B$, and let $A$ be any minimal extensional $\Sigma$ algebra. Let $\text{Eqn}_A(\Sigma)^1$ be the *ground first-order equational theory* of $A$,

$$\text{Eqn}_A(\Sigma)^1 = \{ t = t' \mid \tau \in B \text{ and } t, t' \in T(\Sigma)_\tau \text{ and } A \models t = t' \}.$$  

Then clearly infinitary higher-order equational logic is conservative on $\text{Eqn}_A(\Sigma)^1$ for ground first-order equations. In fact, by induction on the complexity of types, it is easily established that

$$\text{Eqn}_A(\Sigma)^1 \vdash_\omega t = t' \iff A \models t = t'$$

for every ground equation $t = t'$, and thus $A \cong I_{\text{Ext}}(\Sigma, \text{Eqn}_A(\Sigma)^1)$.

(i.b). Similarly, if $\text{Eqn}_A(\Sigma, X)^1$ is the full first-order equational theory of $A$ (where $X_\tau$ is a countably infinite set of variables for each $\tau \in S$) then infinitary higher-order equational logic is conservative on $\text{Eqn}_A(\Sigma, X)^1$ for first-order equations. However, conservativity may not hold for second, or higher-order equations.

(ii). Consider the second-order type structure $S = \{ \text{nat}, (\text{nat} \to \text{nat}) \}$. Define the $S$-typed signature $\Sigma^1$, where

$$\Sigma^1_{\lambda, \text{nat}} = \{ 0 \}, \quad \Sigma^1_{\text{nat}, \text{nat}} = \{ \text{succ} \},$$

$$\Sigma^1_{\lambda, (\text{nat} \to \text{nat})} = \{ \text{eval} \}, \quad \Sigma^1_{(\text{nat} \to \text{nat}), (\text{nat} \to \text{nat})} = \{ f \}.$$  

Define the second-order equational theory $E_1$ to be the set of equations

$$\text{Even}(x) = 0, \quad \text{Odd}(x) = \text{Even}(x), \quad (1, 2)$$

$$f(\text{Even})(x) = 0. \quad (3)$$

Then $E_1 \vdash_\omega f(\text{Even})(0) = 0$, but $E_1 \not\vdash f(\text{Even})(0) = 0$ since $I(\Sigma^1, E_1) \models f(\text{Even})(0) = 0$. So infinitary higher-order equational logic is not conservative on $E_1$ for ground first-order equations.

(iii). Given $S$ as in (ii) above, let $\Sigma^2$ be the $S$-typed signature obtained by deleting the operation symbol $f$ from $\Sigma^1$ in (ii) above. Let $E_2$ be the equational theory obtained by adding to equation (1) in (ii) above the recursion equations

$$\text{Even}(0) = 0, \quad \text{Even}(\text{succ}(x)) = \text{Even}(x). \quad (4, 5)$$

Then $E_2 \vdash_\omega \text{Even}(x) = \text{Odd}(x)$ but $E_2 \not\vdash \text{Even}(x) = \text{Even}(x)$ since in the free model $T_{E_2}(\Sigma^2, X)$ on infinitely many variables of each type we have $T_{E_2}(\Sigma^2, X) \not\models \text{Even}(x) = \text{Odd}(x)$. Thus infinitary higher-order equational logic is not conservative on $E_2$ for first-order equations. However, note that it is conservative on $E_2$ for ground first-order equations. \hfill \square

One aspect of conservativity for higher-order equational logics which is of particular interest is the existence of *normal forms* for higher-order equational proofs. For the finitary higher-order equational calculus, the following class of normal form proofs can be identified.
2.13. Definition. Let \( E \subseteq Eqn(\Sigma, X) \) be any higher-order equational theory. For each type \( \tau \in S \), we define the set of all finitary higher-order equational proofs from \( E \) of type \( \tau \) in **eval normal form**, by induction on the complexity of \( \tau \).

(i). For each basic type \( \tau \in B \) and any terms \( t, t' \in T(\Sigma, X)_\tau \), if \( P \) is a proof of \( t = t' \) from \( E \) using only rules (i)-(iv) of Definition 2.5. (i.e. first-order many-sorted equational logic) then \( P \) is in eval normal form.

(ii). For each product type \( (\sigma \times \tau) \in S \) and any terms \( t, t' \in T(\Sigma, X)_{(\sigma \times \tau)} \), if \( P_1 \) and \( P_2 \) are proofs of \( \text{proj}_1(t) = \text{proj}_1(t') \) and \( \text{proj}_2(t) = \text{proj}_2(t') \) respectively, from \( E \) in eval normal form, then

\[
\frac{P_1 \ P_2}{t = t'}
\]

is a proof of \( t = t' \) from \( E \) in eval normal form.

(iii). For each function type \( (\sigma \rightarrow \tau) \in S \) and any terms \( t, t' \in T(\Sigma, X)_{(\sigma \rightarrow \tau)} \) and any variable \( x \in X_\sigma \) not occurring in \( t \) or \( t' \), if \( P \) is a proof of \( \text{eval}(t, x) = \text{eval}(t', x) \) from \( E \) in eval normal form, then

\[
\frac{P}{t = t'}
\]

is a proof of \( t = t' \) from \( E \) in eval normal form. \( \square \)

Clearly, a finitary proof is in eval normal form when every extensionality or projection rule of each type is used at most once, and occurs at the end of the proof in the manner indicated. That is to say, the proof can be divided into an initial section, consisting of purely equational reasoning, and a final section, where only extensionality and projection inferences are used. A similar concept of eval normal form can be introduced for infinitary higher-order equational proofs. The definition is left as an exercise for the reader.

The existence of eval normal form proofs is of importance for automated reasoning with finitary higher-order equational logic. In particular, it provides a uniform way to reduce the problem of constructing a finitary higher-order equational proof to the problem of constructing a first-order equational proof (for example by term rewriting). This is significant, since there appears to be no obvious way to implement efficiently the finitary extensionality rule in a computational logic.

The existence of eval normal forms is equivalent to the following conservativity property in higher-order equational logic.

2.14. Proposition. Let \( E \subseteq Eqn(\Sigma, X) \) be any higher-order equational theory. The following are equivalent:

(i) finitary higher-order equational logic is conservative on \( E \) for first-order equations;

(ii) for any equation \( e \in Eqn(\Sigma, X) \) (of any order), if

\[
E \vdash \text{eval } e
\]

then there is a finitary proof \( P \) of \( e \) which is in eval normal form.

Proof. (i) \( \Rightarrow \) (ii) By induction on the complexity of \( \tau \). (ii) \( \Rightarrow \) (i) Immediate from Definition 2.13. \( \square \)

By Proposition 2.14, any characterisation of conservativity for the finitary extensionality rule simultaneously characterises the existence of eval normal form proofs.
3 A Topology of Finite Information.

In this section we introduce a topology on higher-order algebras which will be used in Section 4 to characterise conservativity of the ω-extensionality rule over equational logic. The intuition for this topology is that a basic open set (in any type) contains all elements which share the same specific and finite amount of information. Thus the topology is termed a topology of finite information.

The finite information topology can be defined on both extensional and non-extensional algebras. (This fact is important for Section 4.) It is constructed by induction on the complexity of types in such a way that, for extensional algebras, all spaces are homeomorphic with certain spaces of distinguished type which are termed the elementary or hereditarily Horn types. On extensional algebras, the topology has strong separation and countability properties: in particular, it is metric and second countable in every type if every carrier set of basic type is countable (unlike the Kleene Kreisel topology, see Normann [1980]). On every algebra, the projection, pairing, currying and evaluation mappings are continuous in the finite information topology.

We assume the reader to be familiar with the basic concepts of point set topology, such as the definitions of open and closed sets, a basis and subbasis for a topology, and continuous and open mappings between topological spaces. Recall that a homeomorphism between two topological spaces is a continuous open bijection. We let Nbd(a) denote the set of all neighbourhoods of a point a (open sets containing a) inside a given topological space. All topological prerequisites may be found in, for example, Dugundji [1966] or Kelley [1955].

In order to define the finite information topology, it is necessary to be able to form homeomorphic images of open sets in certain types. For this we introduce a collection of names for distinguished homeomorphism operations which will then be assumed to be present in any higher-order signature.

3.1. Definition. Let Σ be an H(B)-typed signature over a type basis B. We say that Σ contains the homeomorphism operators, if, and only if, Σ includes the following families of function symbols,

(i) For each σ, τ, δ ∈ H(B), **currying and uncurrying** function symbols,

\[ cu \in \Sigma((\sigma \times \tau) \rightarrow \delta), (\sigma \rightarrow (\tau \rightarrow \delta)) \] \[ uc \in \Sigma(\sigma \rightarrow (\tau \rightarrow \delta)), ((\sigma \times \tau) \rightarrow \delta). \]

(ii) For each σ, τ, δ, ε ∈ H(B), **generalised currying and inverse generalised currying** function symbols,

\[ gcu \in \Sigma(\epsilon \rightarrow (\sigma \rightarrow \delta)), (\epsilon \rightarrow (\sigma \rightarrow (\tau \rightarrow \delta))) \] \[ gcu^{-1} \in \Sigma((\sigma \rightarrow (\tau \rightarrow \delta)), (\epsilon \rightarrow (\sigma \times \tau) \rightarrow \delta)). \]

(iii) For each σ, τ, δ ∈ H(B), **function-pairing and inverse function-pairing** function symbols,

\[ fp \in \Sigma(\sigma \rightarrow (\tau \times (\sigma \rightarrow \delta))), (\sigma \rightarrow (\tau \times (\tau \times \delta))) \] \[ fp^{-1} \in \Sigma((\sigma \rightarrow (\tau \times \delta)), ((\sigma \rightarrow (\tau \times \delta)) \rightarrow \tau \times (\sigma \rightarrow \delta))). \]

(iv) For each σ, τ, δ, ε ∈ H(B), **generalised function-pairing and inverse generalised function-pairing** function symbols,

\[ gfp \in \Sigma(\epsilon \rightarrow (\sigma \rightarrow (\tau \times (\sigma \rightarrow \delta))), (\epsilon \rightarrow (\sigma \rightarrow (\tau \times (\tau \times \delta)))) \] \[ gfp^{-1} \in \Sigma((\epsilon \rightarrow (\sigma \rightarrow (\tau \times \delta))), (\epsilon \rightarrow (\sigma \rightarrow (\tau \times (\tau \times \delta)))) \rightarrow (\sigma \rightarrow (\tau \times (\sigma \rightarrow \delta))). \]

(v) For each σ, τ, δ, γ ∈ H(B), **right-pairing and inverse right-pairing** function symbols,

\[ rp \in \Sigma(\sigma \rightarrow (\tau \times (\sigma \rightarrow \delta) \rightarrow \gamma)), ((\sigma \times (\tau \times \delta)) \rightarrow \gamma) \] \[ rp^{-1} \in \Sigma((\sigma \times (\tau \times \delta)) \rightarrow \gamma), ((\sigma \times (\sigma \rightarrow \delta) \rightarrow \gamma)). \]
(vi) For each $\sigma, \tau, \delta, \epsilon, \gamma \in H(B)$, generalised right-pairing and inverse generalised right-pairing function symbols,

$$\text{grp} \in \Sigma_{(((\sigma \times \tau) \times \delta) \times \epsilon) \rightarrow \gamma}, \quad \text{grp}^{-1} \in \Sigma_{(((\sigma \times \tau) \times \delta) \times \epsilon) \rightarrow \gamma}. $$

(vii) For each $\sigma, \tau, \delta \in H(B)$, left-bracketing and inverse left-bracketing function symbols,

$$\text{lb} \in \Sigma_{(\sigma \times \tau) \times \delta}, \quad \text{lb}^{-1} \in \Sigma_{((\sigma \times \tau) \times \delta), (\sigma \times \tau \times \delta)}. $$

(viii) For each $\sigma, \tau, \epsilon \in H(B)$, generalised left-bracketing and inverse generalised left-bracketing function symbols,

$$\text{glb} \in \Sigma_{((\sigma \times \tau) \times \epsilon), (\sigma \times \tau \times \epsilon)}, \quad \text{glb}^{-1} \in \Sigma_{(((\sigma \times \tau) \times \delta), (\sigma \times \tau \times \delta)}. $$

(ix) For each $\sigma, \tau \in H(B)$, pairing

$$\langle \cdot, \cdot \rangle \in \Sigma_{\sigma \tau, (\sigma \times \tau)}. $$

In the sequel, we will assume that $\Sigma$ is a fixed, but arbitrarily chosen $H(B)$-typed signature, over a type basis $B$, and that $\Sigma$ contains the homeomorphism operators. Then we can use the operations named by the distinguished homeomorphism operation symbols, in any algebra $A$, to simultaneously define collections of subbasic open sets and the continuous elements in $A$.

3.2. Definition. For each type $\tau \in H(B)$, we define:

(i) the collection of all subbasic open subsets of $A_{\tau}$; and,

(ii) the set $C(A)_{\tau}$ of all continuous elements of $A_{\tau}$,

by induction on the complexity of $\tau$. Then for such $\tau$ as usual the basic open subsets of $A_{\tau}$ are precisely the finite intersections of subbasic open subsets of $A_{\tau}$ and the open subsets of $A_{\tau}$ are precisely the unions of basic open subsets of $A_{\tau}$.

Basis. (i). Consider any basic type $\tau \in B$. A subset $U \subseteq A_{\tau}$ is subbasic open if, and only if, $U$ is a singleton set. Every element of $A_{\tau}$ is continuous, i.e. $C(A)_{\tau} = A_{\tau}$.

Induction Step. (ii). Consider any product type $(\sigma \times \tau) \in H(B)$. A subset $U \subseteq A_{(\sigma \times \tau)}$ is subbasic open if, and only if, $U$ has the form

$$\langle V_{1}, C(A)_{\tau} \rangle$$

for $V_{1} \subseteq A_{\sigma}$ an open subset of $A_{\sigma}$, or

$$\langle C(A)_{\sigma}, V_{2} \rangle$$

for $V_{2} \subseteq A_{\tau}$ an open subset of $A_{\tau}$. For any $a \in A_{(\sigma \times \tau)}$, $a$ is continuous if, and only if, $\text{proj}_{1}^{-1}(a)$ and $\text{proj}_{2}^{-1}(a)$ are continuous.

(iii). Consider any function type $(\sigma \rightarrow \tau) \in H(B)$. We proceed by a subinduction on the complexity of $\tau$.

Subbasis. (a). Suppose $\tau \in B$ is a basic type. A subset $U \subseteq A_{(\sigma \rightarrow \tau)}$ is subbasic open if, and only if, $U$ has the form

$$O_{V, b} = \{ \; a \in A_{(\sigma \rightarrow \tau)} \mid \text{for all } a_{0} \in V, \text{eval}_{A}(a, a_{0}) = b \; \}$$

for $V \subseteq A_{\sigma}$ a basic open subset and $b \in A_{\tau}$. For any $a \in A_{(\sigma \rightarrow \tau)}$, $a$ is continuous if, and only if, for any open subset $U \subseteq A_{\tau}$,

$$a^{-1}(U) = \{ \; b \in A_{\sigma} \mid \text{eval}_{A}(a, b) \in U \; \}$$
is open.

**Subinduction Step.** (b) Suppose $\tau = (\tau_1 \times \tau_2)$ is a product type. A subset $U \subseteq A_{(\sigma \rightarrow (\tau_1 \times \tau_2))}$ is subbasic open if, and only if, $fp_A^{-1}(U)$ is subbasic open. For any $a \in A_{(\sigma \rightarrow (\tau_1 \times \tau_2))}$, $a$ is continuous if, and only if, $fp_A^{-1}(a)$ is continuous.

(c) Suppose $\tau = (\tau_1 \rightarrow \tau_2)$ is a function type. For any $\sigma \in A_{\tau}$, $\sigma \rightarrow \tau$ is a subbasic open subset of $A_{\tau}$ if, and only if, $uc_A(U)$ is subbasic open. For any $a \in A_{(\sigma \rightarrow (\tau_1 \rightarrow \tau_2))}$, $a$ is continuous if, and only if, $uc_A(a)$ is continuous.

**Subbasis.** (c.i) Suppose $\tau_2 \in B$ is a basic type. A subset $U \subseteq A_{(\sigma \rightarrow (\tau_1 \rightarrow \tau_2))}$ is subbasic open if, and only if, $uc_A(U)$ is subbasic open. For any $a \in A_{(\sigma \rightarrow (\tau_1 \rightarrow \tau_2))}$, $a$ is continuous if, and only if, $uc_A(a)$ is continuous.

**Subinduction Step.**

(c.ii) Suppose $\tau_2 = (\delta_1 \times \delta_2)$ is a product type. A subset $U \subseteq A_{(\sigma \rightarrow (\tau_1 \rightarrow (\delta_1 \times \delta_2)))}$ is subbasic open if, and only if, $gfp_A^{-1}(U)$ is subbasic open. For any $a \in A_{(\sigma \rightarrow (\tau_1 \rightarrow (\delta_1 \times \delta_2)))}$, $a$ is continuous if, and only if, $gfp_A^{-1}(a)$ is continuous.

(c.iii) Suppose $\tau_2 = (\delta_1 \rightarrow \delta_2)$ is a function type. A subset $U \subseteq A_{(\sigma \rightarrow (\tau_1 \rightarrow (\delta_1 \rightarrow \delta_2)))}$ is subbasic open if, and only if, $gcu_A^{-1}(U)$ is subbasic open. For any $a \in A_{(\sigma \rightarrow (\tau_1 \rightarrow (\delta_1 \rightarrow \delta_2)))}$, $a$ is continuous if, and only if, $gcu_A^{-1}(a)$ is continuous.

We say that $A$ has **continuous carrier sets** if, and only if, for each type $\tau \in S$, $A_\tau = C(A)_\tau$, i.e. every element of every carrier set is a continuous element. We define the **finite information topology** $FI(A)_\tau$ on $A_\tau$ to be the set of all open subsets of $A_\tau$,

$$FI(A)_\tau = \{ U \subseteq A_\tau \mid U \text{ is open} \}. \quad \square$$

Clearly, the subbasic open subsets of $A$ form a subbasis for a topology, and hence $FI(A)_\tau$ is a well defined topology on $A$, if, and only if, $A$ has continuous carrier sets. Otherwise, if some element $a$ of $A$ is not continuous then there is no subbasic open subset of $A$ which contains $a$.

The homeomorphisms named in Definition 3.1 are associated with certain equational axioms, which must be satisfied in any $\Sigma$ algebra $A$ in order to ensure that these operations really are homeomorphisms. We collect together these equational axioms as follows.

3.3. **Definition.** The **homeomorphism axioms Hom** consist of the following set of equations, for all types $\sigma, \tau, \delta, \epsilon \in H(B)$.

(i) **Currying.** For $x \in X_{(\sigma \rightarrow (\tau \rightarrow \delta))}$, $x' \in X_{((\sigma \times \tau) \rightarrow \delta)}$, $y \in X_\sigma$ and $z \in X_\tau$,

$$cu uc x x \quad (1.a, b), \quad uc cu x' = x' \quad (1.a, b),$$

$$eval uc x y z = eval eval x y z \quad (1.c, d).$$

(ii) **Generalised Currying.** For all $\sigma, \tau, \delta, \epsilon \in H(B)$ and for $x \in X_{(\epsilon \rightarrow (\sigma \rightarrow (\tau \rightarrow \delta)))}$, $x' \in X_{(\epsilon \rightarrow ((\sigma \times \tau) \rightarrow \delta))}$ and $y \in X_\epsilon$,

$$gcu gcu^{-1} x x \quad (2.a, b), \quad gcu^{-1} gcu x' = x' \quad (2.a, b),$$

$$eval gcu^{-1} x y = uc eval x y \quad (2.c, d), \quad eval gcu x y = cu eval x' y \quad (2.c, d),$$

$$gcu^{-1} x = cu(rp uc x) \quad (2.e, f), \quad gcu x' = cu(rp^{-1} uc x') \quad (2.e, f).$$

(iii) **Function-pairing.** For $x \in X_{((\sigma \rightarrow \tau) \times (\sigma \rightarrow \delta))}$, $x' \in X_{(\sigma \rightarrow (\tau \times \delta))}$ and $y \in X_\sigma$,

$$fp^{-1} fp x x \quad (3.a, b), \quad fp fp^{-1} x' = x' \quad (3.a, b),$$

$$eval proj fp^{-1} x' y eval proj^2fp^{-1} x' y \quad (3.c), \quad eval x' y \quad (3.c),$$

14
\[ \text{eval}(fp(x), y) = \langle \text{eval}(\text{proj}^1(x), y), \text{eval}(\text{proj}^2(x), y) \rangle \]  
(3.d).

(v). Generalised Function-pairing. For \( x \in X_{((\sigma \rightarrow (\tau \times x)) \times (\sigma \rightarrow \delta))} \), \( x' \in X_{((\sigma \rightarrow (\tau \times x)) \times \gamma)} \), and \( y \in X_{\sigma} \),

\[
gfp(\text{grp}^{-1}(x')) = x', \quad gfp^{-1}(\text{grp}(x)) = x \tag{4.a,b},
\]

\[
eval(gfp^{-1}(x'), y) = \text{fp}^{-1}(\text{eval}(x', y)), \quad eval(gfp(x), y) = \text{fp}(\text{eval}(x, y)) \tag{4.c,d},
\]

\[
gfp^{-1}(x') = \text{fp}(\langle \text{cu}(\text{proj}^1(\text{fp}^{-1}(uc(x'))), \text{cu}(\text{proj}^2(\text{fp}^{-1}(uc(x')))) \rangle) \tag{4.e},
\]

\[
gfp(x) = \text{cu}(\text{fp}(\langle uc(\text{proj}^1(\text{fp}^{-1}(x'))), uc(\text{proj}^2(\text{fp}^{-1}(x'))) \rangle)) \tag{4.f}.
\]

(v). Right-pairing. For \( x \in X_{((\sigma \times \tau) \times x) \rightarrow \gamma} \), \( x' \in X_{((\sigma \times \tau) \times x) \rightarrow \gamma} \), and \( y \in X_{\sigma} \),

\[
\text{rp}(\text{rp}^{-1}(x')) = x', \quad \text{rp}^{-1}(\text{rp}(x)) = x \tag{5.a,b}.
\]

If \( \gamma \) is a basic type \( \gamma \in B \) then we have

\[
eval(\text{rp}(x), y) = \text{eval}(x, \text{lb}(y)), \quad eval(\text{rp}^{-1}(x'), y) = \text{eval}(x', \text{lb}^{-1}(y)) \tag{5.c,d},
\]

if \( \gamma \) is a product type \( \gamma = (\gamma_1 \times \gamma_2) \) then we have

\[
\text{rp}(x) = \text{fp}(\langle \text{rp}(\text{proj}^1(\text{rp}^{-1}(x'))), \text{rp}(\text{proj}^2(\text{rp}^{-1}(x'))) \rangle) \tag{5.e},
\]

\[
\text{rp}^{-1}(x') = \text{fp}(\langle \text{rp}^{-1}(\text{proj}^1(\text{rp}^{-1}(x'))), \text{rp}^{-1}(\text{proj}^2(\text{rp}^{-1}(x'))) \rangle) \tag{5.f},
\]

and if \( \gamma \) is a function type \( \gamma = (\gamma_1 \rightarrow \gamma_2) \) then we have

\[
\text{rp}(x) = \text{cu}(\text{grp}(uc(x))), \quad \text{rp}^{-1}(x') = \text{cu}(\text{grp}^{-1}(uc(x'))) \tag{5.g,h}.
\]

(vi). Generalised Right-pairing. For \( x \in X_{(((\sigma \times \tau) \times x) \times x) \rightarrow \gamma} \), \( x' \in X_{(((\sigma \times \tau) \times x) \times x) \rightarrow \gamma} \), and \( y \in X_{\sigma} \),

\[
\text{grp}(\text{grp}^{-1}(x')) = x', \quad \text{grp}^{-1}(\text{grp}(x)) = x \tag{6.a,b},
\]

If \( \gamma \) is a basic type \( \gamma \in B \) then we have

\[
eval(\text{grp}(x), y) = \text{eval}(x, \text{glb}(y)), \quad eval(\text{grp}^{-1}(x'), y) = \text{eval}(x', \text{glb}^{-1}(y)) \tag{6.c,d},
\]

if \( \gamma \) is a product type \( \gamma = (\gamma_1 \times \gamma_2) \) then we have

\[
\text{grp}(x) = \text{fp}(\langle \text{grp}(\text{proj}^1(\text{grp}^{-1}(x'))), \text{grp}(\text{proj}^2(\text{grp}^{-1}(x'))) \rangle) \tag{6.e},
\]

\[
\text{grp}^{-1}(x') = \text{fp}(\langle \text{grp}^{-1}(\text{proj}^1(\text{grp}^{-1}(x'))), \text{grp}^{-1}(\text{proj}^2(\text{grp}^{-1}(x'))) \rangle) \tag{6.f},
\]

and if \( \gamma \) is a function type \( \gamma = (\gamma_1 \rightarrow \gamma_2) \) then we have

\[
\text{grp}(x) = \text{cu}(\text{rp}^{-1}(\text{grp}(\text{rp}(uc(x))))), \quad \text{grp}^{-1}(x) = \text{cu}(\text{rp}^{-1}(\text{grp}(\text{rp}(uc(x))))) \tag{6.g,h}.
\]

(vii). Left-bracketing. For \( x \in X_{(\sigma \times \tau) \times \delta} \) and \( x' \in X_{(\sigma \times \tau) \times \delta} \),

\[
\text{lb}(\text{lb}^{-1}(x')) = x', \quad \text{lb}^{-1}(\text{lb}(x)) = x \tag{7.a,b},
\]

\[
\text{lb}(x) = \langle \text{proj}^1(x), \text{proj}^1(\text{proj}^2(x)), \text{proj}^2(\text{proj}^2(x)) \rangle \tag{7.c},
\]

\[
\text{lb}^{-1}(x) = \langle \text{proj}^1(\text{proj}^1(x)), \text{proj}^2(\text{proj}^1(x)), \text{proj}^2(\text{proj}^1(x)) \rangle \tag{7.d}.
\]
(viii). Generalised Left-bracketing. For \( x \in X_{(\sigma \times (\tau \times \delta)) \times \epsilon} \) and \( x' \in X_{((\sigma \times \tau) \times \delta) \times \epsilon} \),
\[ \text{glb} (\text{glb}^{-1}(x')) = x', \quad \text{glb}^{-1}(\text{glb}(x)) = x \tag{8.a,b} \]
\[ \text{glb}(x) = \langle \text{proj}^1(x), \text{proj}^2(x) \rangle, \quad \text{glb}^{-1}(x') = \langle \text{proj}^{-1}^1(x'), \text{proj}^2(x') \rangle \tag{8.c,d} \]

(ix). Pairing. For any \( x \in X_{(\sigma \times \tau)} \) and \( y \in X_\sigma \) and \( z \in X_\tau \),
\[ x = \langle \text{proj}^1(x), \text{proj}^2(x) \rangle \tag{9.a} \]
\[ y = \text{proj}^1((y,z)), \quad z = \text{proj}^2((y,z)) \tag{9.b,c} \]

\[ \square \]

In the sequel, we let \( A \in \text{Alg}(\Sigma, \text{Hom}) \) denote a fixed, but arbitrarily chosen \( \Sigma \) algebra which satisfies the homeomorphism axioms \( \text{Hom} \) and has continuous carrier sets. (Note that \( A \) is not necessarily extensional.) Then the finite information topology \( \text{FI}(A) \) is well defined for \( A \). We will show that all the homeomorphism operations named in \( \Sigma \) are continuous open mappings when interpreted in \( A \). First we establish a number of technical facts.

3.4. Proposition. For any types \( \sigma, \tau \in H(B) \), \( A_{(\sigma \times \tau)} = \langle A_\sigma, A_\tau \rangle_A \).

Proof. \( A_{(\sigma \times \tau)} = \langle \text{proj}^1_A(A_{(\sigma \times \tau)}), \text{proj}^2_A(A_{(\sigma \times \tau)}) \rangle_A = \langle A_\sigma, A_\tau \rangle_A \). \[ \square \]

3.5. Proposition. For any types \( \sigma, \tau, \delta \in H(B) \), and any open sets \( V_1 \in \text{FI}(A)_{\sigma} \), \( V_2 \in \text{FI}(A)_{\tau} \), and \( V_3 \in \text{FI}(A)_{\delta} \):
(i) \( \text{lb}_A(\langle V_1, A_{(\tau \times \delta)} \rangle_A) \),
(ii) \( \text{lb}_A(\langle A_\sigma, \langle V_2, A_\delta \rangle \rangle_A) \),
(iii) \( \text{lb}_A(\langle A_\sigma, \langle A_\tau, V_3 \rangle \rangle_A) \),
are all subbasic open sets.

Proof. (i) By Proposition 3.4,
\[ \text{lb}_A(\langle V_1, A_{(\tau \times \delta)} \rangle_A) = \text{lb}_A(\langle V_1, \langle A_\tau, A_\delta \rangle_A \rangle_A) = \langle \langle V_1, A_\tau \rangle_A, A_\delta \rangle_A \]
which is subbasic open. The proofs of (ii) and (iii) are similar. \[ \square \]

3.6. Proposition. Let \( \sigma, \tau \in H(B) \) be any types.
(i) For any open sets \( V_1 \in \text{FI}(A)_{\sigma} \) and \( V_2 \in \text{FI}(A)_{\tau} \),
\[ \langle V_1, V_2 \rangle_A \in \text{FI}(A)_{(\sigma \times \tau)} \]
(ii) For any open set \( U \in \text{FI}(A)_{(\sigma \times \tau)} \) and \( V_1 \in \text{FI}(A)_{\sigma} \) and \( V_2 \in \text{FI}(A)_{\tau} \),
\[ U \cap \langle V_1, A_\tau \rangle_A = \langle \text{proj}^1_A(U) \cap V_1, \text{proj}^2_A(U) \cap A_\tau \rangle_A \tag{a} \]
and
\[ U \cap \langle A_\sigma, V_2 \rangle_A = \langle \text{proj}^1_A(U) \cap A_\sigma, \text{proj}^2_A(U) \cap V_2 \rangle_A \tag{b} \]
(iii) For any open set \( U \in \text{FI}(A)_{(\sigma \times \tau)} \),
\[ \text{proj}^1_A(U) \in \text{FI}(A)_{\sigma}, \quad \text{proj}^2_A(U) \in \text{FI}(A)_{\tau} \].
Proof. See Appendix. \[\square\]

3.7. Proposition. For any types \(\sigma, \tau, \delta \in H(B)\) and any open sets \(V_1 \in FI(A)_\sigma, V_2 \in FI(A)_\tau, V_3 \in FI(A)_\delta\) and \(U \in FI(A)_{(\sigma \times (\tau \times \delta))}\):

(i) \(\text{lb}_A(V_1, A_{(\tau \times \delta)}) \cap U) = \text{lb}_A(V_1, A_{(\tau \times \delta)}) \cap \text{lb}_A(U);\)

(ii) \(\text{lb}_A(A_{\sigma}, (V_2, A_{\delta})) \cap U) = \text{lb}_A(A_{\sigma}, (V_2, A_{\delta})) \cap \text{lb}_A(U);\)

(iii) \(\text{lb}_A(A_{\sigma}, (A_\tau, V_3)) \cap U) = \text{lb}_A(A_{\sigma}, (A_\tau, V_3)) \cap \text{lb}_A(U).\)

Proof. See Appendix. \[\square\]

3.8. Proposition. For any types \(\sigma, \tau, \delta \in H(B)\) and any open set \(U \in FI(A)_{(\tau \times \delta)},\)

\[\langle A_{\sigma}, U \rangle_A = \langle A_{\sigma}, \langle \text{proj}_A^1(U), A_{\delta} \rangle_A \cap \langle A_{\sigma}, \langle \text{proj}_A^2(U) \rangle_A \rangle_A.\]

Proof. By Proposition 3.6.(ii.b),

\[\langle A_{\sigma}, \langle \text{proj}_A^1(U), A_{\delta} \rangle_A \cap \langle A_{\sigma}, \langle \text{proj}_A^2(U) \rangle_A \rangle_A = \langle A_{\sigma}, \langle \text{proj}_A^1(U), A_{\delta} \rangle_A \cap \langle A_{\sigma}, \langle \text{proj}_A^2(U) \rangle_A \rangle_A = \langle A_{\sigma}, U \rangle_A.\]

3.9. Lemma. Let \(\sigma, \tau, \delta, \epsilon \in H(B)\) be any types. The operations

\[\text{lb}_A : A_{(\sigma \times (\tau \times \delta))} \rightarrow A_{(\sigma \times (\tau \times \delta))}, \quad \text{lb}_A^{-1} : A_{(\sigma \times (\tau \times \delta))} \rightarrow A_{(\sigma \times (\tau \times \delta))}\]

\[\text{glb}_A : A_{(((\sigma \times (\tau \times \delta)) \times \epsilon))} \rightarrow A_{(((\sigma \times (\tau \times \delta)) \times \epsilon))}, \quad \text{glb}_A^{-1} : A_{(((\sigma \times (\tau \times \delta)) \times \epsilon))} \rightarrow A_{(((\sigma \times (\tau \times \delta)) \times \epsilon))}\]

all preserve basic open sets.

Proof. (i). We prove that

\[\text{lb}_A : A_{(\sigma \times (\tau \times \delta))} \rightarrow A_{(\sigma \times (\tau \times \delta))}\]

preserves basic open sets as follows. Consider any basic open set \(U \subseteq A_{(\sigma \times (\tau \times \delta))}\). For some \(m \geq n\), by Proposition 3.8, there exist subbasic open sets \(W_1, \ldots, W_m\) with

\[U = W_1 \cap \ldots \cap W_m,\]

and for each \(1 \leq i \leq m\), either

\[W_i = \langle W^1_i, A_{\tau}, A_{\delta} \rangle_A,\]

or

\[W_i = \langle A_{\sigma}, W^2_i, A_{\delta} \rangle_A,\]

or

\[W_i = \langle A_{\sigma}, A_{\tau}, W^3_i \rangle_A,\]

for some open sets \(W^1_i \in FI(A)_\sigma, W^2_i \in FI(A)_\tau\) and \(W^3_i \in FI(A)_\delta\). By Proposition 3.7,

\[\text{lb}_A(U) = \text{lb}_A(W^1_1) \cap \ldots \cap \text{lb}_A(W^1_m).\]
By Proposition 3.5, for each \(1 \leq i \leq m\), \(lb_A(W_i)\) is subbasic open, and so \(lb_A(U)\) is basic open.

The proofs that \(lb^{-1}\), \(glb\) and \(glb^{-1}\) preserve basic open sets are obtained similarly by modifying Proposition 3.7 appropriately.

The continuity and openness of some of the mappings named in Definition 3.1 will be used to prove the continuity and openness of others. Thus we prove the following three (apparently similar) theorems in the specific order they appear, and using slightly different methods in each case.

3.10. Theorem. For any types \(\sigma, \tau, \delta, \epsilon, \gamma \in H(B)\), the following operations are continuous open mappings:

(i) Generalised function-pairing: \(gfp_A : A_{(\sigma \rightarrow (\tau \rightarrow \delta \times (\tau \rightarrow \epsilon)))} \rightarrow A_{(\sigma \rightarrow (\tau \rightarrow (\delta \times \epsilon)))}\),

(ii) Generalised currying: \(gcu_A : A_{(\sigma \rightarrow ((\tau \times \delta) \rightarrow \epsilon))} \rightarrow A_{(\sigma \rightarrow ((\tau \rightarrow \delta) \rightarrow \epsilon))}\),

(iii) Left and right projection: \(proj_A^1 : A_{((\sigma \times \tau) \rightarrow \delta)} \rightarrow A_\sigma, proj_A^2 : A_{((\sigma \times \tau) \rightarrow \delta)} \rightarrow A_\tau\),

(iv) Pairing: \(\langle \cdot, \cdot \rangle_A : A_\sigma \times A_\tau \rightarrow A_{(\sigma \times \tau)}\),

(v) Left-bracketing: \(lb_A : A_{(\sigma \times (\tau \times \delta))} \rightarrow A_{((\sigma \times \tau) \times \delta)}\),

(vi) Generalised left-bracketing: \(glb_A : A_{(((\sigma \times \tau) \times \delta) \times \epsilon)} \rightarrow A_{(((\sigma \times \tau) \times \delta) \times \epsilon)}\),

(vii) Function-pairing: \(fp_A : A_{(((\sigma \times \tau) \times (\sigma \rightarrow \delta))} \rightarrow A_{(\sigma \rightarrow (\tau \times \delta))}\).

Proof. See Appendix.

3.11. Theorem. For any types \(\sigma, \tau, \delta, \epsilon, \gamma \in H(B)\), the following operations are continuous open mappings:

(i) Currying: \(cu_A : A_{((\sigma \times \tau) \rightarrow \gamma)} \rightarrow A_{(\sigma \rightarrow (\tau \rightarrow \gamma))}\),

(ii) Right-pairing: \(rp_A : A_{(((\sigma \times \tau) \times \delta) \rightarrow \gamma)} \rightarrow A_{((\sigma \times (\tau \times \delta)) \rightarrow \gamma)}\),

(iii) Generalised right-pairing: \(grp_A : A_{(((\sigma \times \tau) \times \delta) \times \epsilon) \rightarrow \gamma)} \rightarrow A_{(((\sigma \times (\tau \times \delta)) \times \epsilon) \rightarrow \gamma)}\).

Proof. See Appendix.

3.12. Theorem. For any types \(\sigma, \tau \in H(B)\), the evaluation mapping

\[ eval_A : A_{(\sigma \rightarrow \tau)} \times A_\sigma \rightarrow A_\tau \]

is continuous.

Proof. We prove the result by induction on the complexity of \(\tau\).

Basis. (i). Suppose \(\tau \in B\) is a basic type. Consider any subbasic open set \(\{c\} \in FI(\sigma)_\tau\) for \(c \in A_\tau\). We must show that

\[ eval_A^{-1}(\{c\}) = \{(a, b) \in A_{(\sigma \rightarrow \tau)} \times A_\sigma \mid eval_A(a, b) = c\} \]

is open. So consider any \(a \in A_{(\sigma \rightarrow \tau)}\) and \(b \in A_\sigma\) such that \(eval_A(a, b) = c\). Since \(a\) is continuous then \(a^{-1}(\{c\})\) is open and \(b \in a^{-1}(\{c\})\). So there exists a basic open set \(U_b \in FI(\sigma)_\sigma\) with \(U_b \subseteq a^{-1}(\{c\})\) and \(b \in U_b\) and for all \(b' \in U_b\),

\[ eval_A(a, b') = eval_A(a, b) = c. \]

Consider the subbasic open set \(O_{U_b, c} \in FI(\sigma)_{(\sigma \rightarrow \tau)}\). Then

\[ (a, b) \in O_{U_b, c} \times A_\sigma \]

is open.
for some

Thus

\[ \bigcup_{(a,b) \in \text{eval}^{-1}_A(\{ \{ c \} \})} O_{U_{b,c}} \subseteq \text{eval}^{-1}_A(\{ \{ c \} \}). \]

and since \( c \) was arbitrarily chosen then \( \text{eval}_A \) is continuous.

**Induction Step.** (ii). Suppose \( \tau = (\tau_1 \times \tau_2) \) is a product type.

Consider any subbasic open set \( \langle V_1, C(A)_{\tau_1} \rangle_A = \langle V_1, A_{\tau_1} \rangle_A \), for open \( V_1 \subseteq \text{FI}(A)_{\tau_1} \). By the induction hypothesis \( \text{eval}_A : A_{(\sigma \to \tau_1)} \times A_\sigma \to A_{\tau_1} \) is continuous. So

\[ U_1 = \text{eval}^{-1}_A(V_1) \]

is open. Now for some indexing set \( I \),

\[ U_1 = \bigcup_{i \in I} W_{i,1} \times W_{i,2}, \]

where for each \( i \in I \), \( W_{i,1} \in \text{FI}(A)_{(\sigma \to \tau_1)} \) and \( W_{i,2} \in \text{FI}(A)_\sigma \) are basic open sets. For each \( i \in I \), define

\[ Y_{i,1} = \text{fp}_A(\langle W_{i,1}, A_{(\sigma \to \tau_2)} \rangle_A), \quad Y_{i,2} = W_{i,2}. \]

Then \( Y_{i,1} \in \text{FI}(A)_{(\sigma \to (\tau_1 \times \tau_2))} \) by Proposition 3.6.(i) and Theorem 3.10.(vii) above and \( Y_{i,2} \in \text{FI}(A)_\sigma \). Let

\[ Y = \bigcup_{i \in I} Y_{i,1} \times Y_{i,2}. \]

We need only show that

\[ Y = \text{eval}^{-1}_A(\langle V_1, C(A)_{\tau_1} \rangle_A). \quad (1) \]

Now

\[ (a, b) \in \text{eval}^{-1}_A(\langle V_1, C(A)_{\tau_1} \rangle_A) \iff \text{eval}_A(a, b) \in \langle V_1, C(A)_{\tau_1} \rangle_A \]

\[ \iff \text{proj}_A^1(\text{eval}_A(a, b)) \in V_1 \text{ and } \text{proj}_A^2(\text{eval}_A(a, b)) \in A_{\tau_1} \]

\[ \iff \text{eval}_A(\text{proj}_A^1(\text{fp}_A^{-1}(a), b)) \in V_1 \text{ and } \text{eval}_A(\text{proj}_A^2(\text{fp}_A^{-1}(a), b)) \in A_{\tau_1} \]

by equation 3.3.(3.c),

\[ (\text{proj}_A^1(\text{fp}_A^{-1}(a)), b) \in W_{i,1} \times W_{i,2} \text{ and } (\text{proj}_A^2(\text{fp}_A^{-1}(a)), b) \in A_{(\sigma \to \tau_2)} \times A_\sigma \]

for some \( i \in I \),

\[ \iff \text{fp}_A(\langle \text{proj}_A^1(\text{fp}_A^{-1}(a)), \text{proj}_A^2(\text{fp}_A^{-1}(a)) \rangle_A) \in Y_{i,1} \text{ and } b \in Y_{i,2} \]

for some \( i \in I \),

\[ \iff a \in Y_{i,1} \text{ and } b \in Y_{i,2} \iff (a, b) \in Y. \]

Thus (1) holds.

Similarly, for any subbasic open set \( \langle C(A)_\sigma, V_2 \rangle_A \) for open \( V_2 \in \text{FI}(A)_{\tau_2} \),

\[ \text{eval}^{-1}_A(\langle C(A)_\sigma, V_2 \rangle_A) \]

is open. It follows that \( \text{eval}_A \) is continuous.

(iii). Suppose that \( \tau = (\tau_1 \to \tau_2) \) is a function type. We proceed by subinduction on the complexity of \( \tau_2 \).

19
Subbasis. (iii.a). Suppose that \( \tau_2 \in B \) is a basic type. Consider any subbasic open set \( O_{U, c} \in FI(A)_{(\tau_1 \rightarrow \tau_2)} \) and any \( a \in A_{(\sigma \rightarrow (\tau_1 \rightarrow \tau_2))} \). Now by assumption \( a \) is continuous, so
\[
V^a = a^{-1}(O_{U, c}) \in FI(A)_\sigma.
\]

Thus by Proposition 3.6.(i),
\[
O_{(V^a, U)}_{A, c} \in FI(A)_{((\sigma \times \tau_1) \rightarrow \tau_2)}
\]
is subbasic open. Then by Theorem 3.11.(i), (continuity of \( uc_A \)),
\[
W^a = cu_A(O_{(V^a, U)}_{A, c}) \in FI(A)_{(\sigma \rightarrow (\tau_1 \rightarrow \tau_2))}.
\]
Let
\[
Y = \cup_{a \in A_{(\sigma \rightarrow (\tau_1 \rightarrow \tau_2)}} W^a \times V^a,
\]
then we need only show that
\[
Y = eval^{-1}_A(O_{U, c}).
\]
Consider any \( a, b \) such that \( eval_A(a, b) \in O_{U, c} \). Then \( b \in V^a \), and for any \( b' \in V^a \), \( eval_A(a, b') \in O_{U, c} \). So for any \( b' \in V^a \) and \( a_0 \in U \),
\[
eval_A(eval_A(a, b'), a_0) = c,
\]
and so
\[
uc_A(a) \in O_{(V^a, U)}_{A, c},
\]
and hence \( a \in W^a \). Thus \( (a, b) \in Y \), and since \( a \) and \( b \) were arbitrarily chosen then \( eval^{-1}_A(O_{U, c}) \subseteq Y \).

Conversely, consider any \( a, b \) such that \( (a, b) \in Y \). Then for some \( a_i \in A_{(\sigma \rightarrow (\tau_1 \rightarrow \tau_2))} \), \( a \in W^{a_i} \) and \( b \in V^{a_i} = a_i^{-1}(O_{U, c}) \). So
\[
a \in cu_A(O_{(V^{a_i}, U)}_{A, c}),
\]
and hence for some \( a_j \in O_{(V^{a_i}, U)}_{A, c} \), \( uc_A(a) = a_j \). Now for any \( a_0 \in V^{a_i} \) and \( a_1 \in U \),
\[
eval_A(a_j, \langle a_0, a_1 \rangle_A) = eval_A(uc_A(a), \langle a_0, a_1 \rangle_A)
\]
\[
= eval_A(eval_A(a, a_0), a_1) = c.
\]
So \( eval_A(a, b) \in O_{U, c} \), and since \( a \) and \( b \) were arbitrarily chosen then \( Y \subseteq eval^{-1}_A(O_{U, c}) \). Since \( O_{U, c} \) was arbitrarily chosen then \( eval_A \) is continuous.

Subinduction Step. (iii.b) Suppose that \( \tau_2 = (\delta_1 \times \delta_2) \) is a product type. Consider any subbasic open set \( U \in FI(A)_{(\tau_1 \rightarrow (\delta_1 \times \delta_2)}) \). Then by Definition 2.2,
\[
fp^{-1}_A(U) \in FI(A)_{((\tau_1 \rightarrow \delta_1) \times (\tau_1 \rightarrow \delta_2))}.
\]
So either
\[
fp^{-1}_A(U) = \langle V_1, A_{(\tau_1 \rightarrow \delta_1)} \rangle_A,
\]
for some open \( V_1 \in FI(A)_{(\tau_1 \rightarrow \delta_1)} \), or
\[
fp^{-1}_A(U) = \langle A_{(\tau_1 \rightarrow \delta_1)}, V_2 \rangle_A,
\]
for some open \( V_2 \in FI(A)_{(\tau_1 \rightarrow \delta_2)} \). Suppose that (1) holds. Then by the subinduction hypothesis,
\[
eval_A : A_{(\sigma \rightarrow (\tau_1 \rightarrow \delta_1))} \times A_\sigma \rightarrow A_{(\tau_1 \rightarrow \delta_1)}
\]

20
is continuous, and so \( \text{eval}_A(\text{proj}_A^1(\text{fp}_A^{-1}(U))) \) is open. Hence for some set \( I \),

\[
\text{eval}_A(\text{proj}_A^1(\text{fp}_A^{-1}(U))) = \bigcup_{i \in I} W_{i,1} \times W_{i,2},
\]

where for each \( i \in I \), \( W_{i,1} \in FI(A)_{(\sigma \rightarrow (\tau_1 \rightarrow \delta_1))} \) and \( W_{i,2} \in FI(A)_{\sigma} \). For each \( i \in I \), define

\[
Y_{i,1} = \text{gfp}_A(\text{fp}_A(\langle W_{i,1}, A_{(\sigma \rightarrow (\tau_1 \rightarrow \delta_1))} \rangle_A))
\]

and

\[
Y_{i,2} = W_{i,2}.
\]

Then by Theorem 3.10.(i) and (vii) (openness of \( \text{gfp}_A \) and \( \text{fp}_A \)), for each \( i \in I \),

\[
Y_{i,1} \in FI(A)_{(\sigma \rightarrow (\tau_1 \rightarrow (\delta_1 \times \delta_2)))}
\]

and \( Y_{i,2} \in FI(A)_{\sigma} \). Let

\[
Y = \bigcup_{i \in I} Y_{i,1} \times Y_{i,2},
\]

then we need only show that \( Y = \text{eval}^{-1}_A(U) \).

Now for any \( a \) and \( b \), \((a, b) \in Y \) if, and only if, for some \( i \in I \),

\[
a \in Y_{i,1} \text{ and } b \in Y_{i,2} \iff
\]

for some \( i \in I \)

\[
\text{proj}_A^1(\text{fp}_A^{-1}(\text{gfp}_A^{-1}(a))) \in W_{i,1} \text{ and } \text{proj}_A^2(\text{fp}_A^{-1}(\text{gfp}_A^{-1}(a))) \in A_{(\sigma \rightarrow (\tau_1 \rightarrow \delta_2))} \text{ and } b \in W_{i,2}
\]

\[
\iff \text{eval}_A(\text{proj}_A^1(\text{fp}_A^{-1}(\text{gfp}_A^{-1}(a))), b) \in \text{proj}_A^1(\text{fp}_A^{-1}(U))
\]

and

\[
\text{eval}_A(\text{proj}_A^2(\text{fp}_A^{-1}(\text{gfp}_A^{-1}(a))), b) \in \text{proj}_A^2(\text{fp}_A^{-1}(U))
\]

\[
\iff (\text{eval}_A(\text{proj}_A^1(\text{fp}_A^{-1}(\text{gfp}_A^{-1}(a))), b), \text{eval}_A(\text{proj}_A^2(\text{fp}_A^{-1}(\text{gfp}_A^{-1}(a))), b))_A
\]

\[
\in (\text{proj}_A^1(\text{fp}_A^{-1}(U)), \text{proj}_A^2(\text{fp}_A^{-1}(U)))_A \iff
\]

\[
\text{eval}_A(\text{gfp}_A^{-1}(a), b) \in \text{fp}_A^{-1}(U) \iff
\]

\[
\text{fp}_A(\text{eval}_A(\text{gfp}_A^{-1}(a), b)) \in U \iff \text{eval}_A(a, b) \in U.
\]

Similarly, if (2) holds then \( \text{eval}^{-1}_A(U) \) is open. Since \( U \) was arbitrarily chosen, then \( \text{eval}_A \) is continuous.

(iii.c) Suppose \( \tau_2 = (\delta_1 \rightarrow \delta_2) \) is a function type. Consider any subbasic open set

\[
U \in FI(A)_{(\tau_1 \rightarrow (\delta_1 \times \delta_2))}.
\]

By Theorem 3.11.(i) above (continuity of \( cu_A \)), \( cu_A(U) \in FI(A)_{((\tau_1 x \delta_1) \rightarrow \delta_2)} \) is open. By the subinduction hypothesis,

\[
\text{eval}_A : A_{(\sigma \rightarrow ((\tau_1 x \delta_1) \rightarrow \delta_2))} \times A_{\sigma} \rightarrow A_{((\tau_1 x \delta_1) \rightarrow \delta_2)}
\]

is continuous. So

\[
\text{eval}^{-1}_A(\text{cu}_A(U))
\]

is open. Hence for some set \( I \),

\[
\text{eval}^{-1}_A(\text{cu}_A(U)) = \bigcup_{i \in I} W_{i,1} \times W_{i,2},
\]

21
where for each \( i \in I \), \( W_{i,1} \in FI(A)_{(\sigma \rightarrow (\tau_1 \times \delta_1) \rightarrow \delta_2)} \) and \( W_{i,2} \in FI(A)_\sigma \). For each \( i \in I \), define
\[
Y_{i,1} = gc u_A(W_{i,1}), \quad Y_{i,2} = W_{i,2}.
\]
Then by Theorem 3.10.(ii), above (openness of \( gc u_A \)), for each \( i \in I \),
\[
Y_{i,1} \in FI(A)_{(\sigma \rightarrow (\tau_1 \rightarrow (\delta_1 \rightarrow \delta_2)))}, \quad Y_{i,2} \in FI(A)_\sigma.
\]
Let
\[
Y = \cup_{i \in I} Y_{i,1} \times i, 2,
\]
then we need only show that
\[
Y = eval_A^{-1}(U).
\]
Now for any \( a \) and \( b \), \((a, b) \in Y \) if, and only if, for some \( i \in I \),
\[
a \in Y_{i,1} \quad \text{and} \quad b \in Y_{i,2} \quad \Leftrightarrow \quad eval_A(gc u_A^{-1}(a), b) \in uc_A(U) \quad \text{and} \quad b \in W_{i,2} \quad \Leftrightarrow \quad eval_A(a, b) \in U.
\]
Since \( U \) was arbitrarily chosen then \( eval_A \) is continuous.

We conclude this section with a brief summary of some of the basic properties of the finite information topology. First, under the assumption of extensionality, we can observe that every space of type \( \tau \) is homeomorphic with some space of distinguished type. These types can be defined inductively as follows.

3.13. Definition. Let \( B \) be a type basis. For each \( n \geq 0 \) we define the set \( E_n(B) \subseteq H(B) \) of all 

(i) \( E_0 = B \).

(ii) For any \( n \geq 0 \) and elementary type \( \sigma \in E_n(B) \) and any basic type \( \beta \in B \),
\[
(\sigma \rightarrow \beta) \in E_{n+1}(B).
\]

(iii) For any \( m, n \geq 0 \) and any elementary types \( \tau \in E_m \) and \( \sigma \in E_n \), if \( k = sup\{ m, n \} \) then
\[
(\sigma \times \tau) \in E_k(B).
\]

We let \( E(B) = \cup_{n \in \omega} E_n(B) \).

It is easily verified that if \( \tau \in E_n(B) \) then \( \tau \) is a type of order \( n \). Following the Curry Howard correspondence between propositions and types, the elementary types are also termed the hereditarily Horn types (interpreting basic types as propositional variables, \( \rightarrow \) as implication and \( \times \) as conjunction).

3.14. Proposition. If \( A \) is extensional then for every type \( \tau \in S \) there exists an elementary type \( n(\tau) \in E(B) \) such that \( FI(A)_\tau \) is homeomorphic with \( FI(A)_{n(\tau)} \).

Proof. By induction on the complexity of types using Theorems 3.10 and 3.11.

In order to investigate separation properties, we require the following fact.

3.15. Proposition. For any types \( \sigma, \tau \in S \) and any closed set \( U \in FI(A)_{(\sigma \times \tau)} \), \( proj_A^1(U) \) and \( proj_A^2(U) \) are closed.
Proof. Consider any $\sigma, \tau \in S$ and closed $U \in FI(A)_{(\sigma \times \tau)}$. It is a routine exercise to show that for any $V_1 \subseteq A_\sigma$ and $V_2 \subseteq A_\tau$,

$$cl(\langle V_1, V_2 \rangle_A) = \langle cl(V_1), cl(V_2) \rangle_A,$$

from which the result follows. \qed

Now we can consider the separation and countability properties of the finite information topology. When $A$ is an extensional algebra these properties are quite strong.

3.16. Theorem.
(i) For each type $\tau \in S$, the topology $FI(A)_\tau$ is regular.
(ii) If for each basic type $\tau \in B$ the set $A_\tau$ is countable, then for each type $\tau \in S$, the topology $FI(A)_\tau$ is second countable.
(iii) If $A$ is extensional then for each type $\tau \in S$, the topology $FI(A)_\tau$ is Hausdorff.

Proof. By induction on the complexity of types.

Basis. Consider any basic type $\tau \in B$.
(i) By definition, $FI(A)_\tau = P(A_\tau)$ is regular.
(ii) By assumption, $A_\tau$ is countable, so clearly $FI(A)_\tau = P(A_\tau)$ is second countable.
(iii) Clearly $FI(A)_\tau = P(A_\tau)$ is Hausdorff.

Induction Step. Consider any product type $(\sigma \times \tau) \in S$.
(i) Consider any $a \in A_{(\sigma \times \tau)}$ and closed subset $U \in FI(A)_{(\sigma \times \tau)}$ with $a \notin U$. Then either

$$proj^1_A(a) \notin proj^1_A(U) \tag{1},$$

or

$$proj^2_A(a) \notin proj^2_A(U) \tag{2}.$$ 

Suppose that (1) holds. Then by the induction hypothesis $FI(A)_\omega$ is regular and by Proposition 3.15, $proj^1_A(U)$ is closed. So there exist open $V, W \in FI(A)_\sigma$ with $proj^1_A(a) \in V$ and $proj^1_A(U) \subseteq W$ and

$$V \cap W = \emptyset.$$ 

So $a \in \langle V, A_\tau \rangle_A$, $U \subseteq \langle W, A_\tau \rangle_A$ and

$$\langle V, A_\tau \rangle_A \cap \langle W, A_\tau \rangle_A = \emptyset.$$ 

Also $\langle V, A_\tau \rangle_A$ and $\langle W, A_\tau \rangle_A$ are subbasic open. Similarly, if (2) holds then there exist open $V, W \in FI(A)_\tau$ with $a \in \langle A_\sigma, V \rangle_A$, $U \subseteq \langle A_\sigma, W \rangle_A$ and

$$\langle A_\sigma, V \rangle_A \cap \langle A_\sigma, W \rangle_A = \emptyset.$$ 

Since $a$ and $U$ were arbitrarily chosen then $FI(A)_{(\sigma \times \tau)}$ is regular.

(ii) By the induction hypothesis $FI(A)_\sigma$ and $FI(A)_\tau$ are second countable, i.e. have countable bases $B_\sigma$ and $B_\tau$. So

$$\{ \langle V_1, V_2 \rangle_A \mid V_1 \in B_\sigma, V_2 \in B_\tau \}$$

is a countable basis for $FI(A)_{(\sigma \times \tau)}$.

(iii) By the induction hypothesis, $FI(A)_\sigma$ and $FI(A)_\tau$ are Hausdorff. Now consider any $a, b \in A_{(\sigma \times \tau)}$ and suppose that $a \neq b$. By assumption $A$ is extensional, so either

$$proj^1_A(a) \neq proj^1_A(b) \tag{1},$$

23
or
\[ \text{proj}_2^A(a) \neq \text{proj}_2^A(b) \]  \( (2) \).

Suppose that (1) holds. Then since \( FI(A)_\sigma \) is Hausdorff, there exist neighbourhoods \( U_a \in Nbd(\text{proj}_2^A(a)) \) and \( U_b \in Nbd(\text{proj}_2^A(b)) \) such that
\[ U_a \cap U_b = \emptyset. \]

So \( \langle U_a, A_\tau \rangle_A \) and \( \langle U_b, A_\tau \rangle_A \) are disjoint neighbourhoods of \( a \) and \( b \) respectively. Similarly if (2) holds then \( a \) and \( b \) have disjoint neighbourhoods. Thus \( FI(A)_{(\sigma \times \tau)} \) is Hausdorff.

Consider any function type \((\sigma \rightarrow \tau) \in S\). Then we prove the result by a subinduction on the complexity of \( \tau \).

**Subbasis.** Suppose that \( \tau \in B \) is a basic type.

(i) To show that \( FI(A)_{(\sigma \rightarrow \tau)} \) is regular, consider any \( a \in A_{(\sigma \rightarrow \tau)} \) and closed subset \( U \subseteq A_{(\sigma \rightarrow \tau)} \) such that \( a \notin U \). Now for some open set \( V \in FI(A)_{(\sigma \rightarrow \tau)} \), \( U = cl(V) \) and since \( a \notin cl(V) \) then for some open neighbourhood \( Y \in FI(A)_{(\sigma \rightarrow \tau)} \) of \( a \),
\[ Y \cap V = \emptyset. \]

So for some basic open neighbourhood \( W \in FI(A)_{(\sigma \rightarrow \tau)} \) of \( a \),
\[ W \cap V = \emptyset. \]

Then for any \( b \in cl(V) \), \( b \notin W \), for if \( b \in W \) then \( W \) is an open neighbourhood of \( b \) which is disjoint from \( V \) contradicting the fact that \( b \) is adherent to \( V \). Hence
\[ W \cap cl(V) = \emptyset. \]

We construct an open set \( X \in FI(A)_{(\sigma \rightarrow \tau)} \) such that
\[ cl(V) \subseteq X \quad \text{and} \quad W \cap X = \emptyset. \]

Now for some \( m \geq 1 \) and basic open \( Y_1, \ldots, Y_m \in FI(A)_\sigma \) and \( a_1, \ldots, a_m \in A_\tau \),
\[ W = O_{Y_1, a_1} \cap \ldots \cap O_{Y_m, a_m}. \]

Consider any \( b \in cl(V) \). Then \( b \notin W \) and so for some \( 1 \leq i(b) \leq m \),
\[ b \notin O_{Y_{i(b), a_{i(b)}}}. \]

Now \( b \) is continuous. So \( b^{-1}(\{ a_{i(b)} \}) \) is open. Hence
\[ Y_{i(b)} \nsubseteq b^{-1}(\{ a_{i(b)} \}). \]

So for some \( y_{i(b)} \in Y_{i(b)} \),
\[ \text{eval}_A(b, y_{i(b)}) \neq a_{i(b)}. \]

Let \( c_{i(b)} = \text{eval}_A(b, y_{i(b)}) \) and consider the open set
\[ b^{-1}(\{ c_{i(b)} \}). \]

Then \( y_{i(b)} \in b^{-1}(\{ c_{i(b)} \}) \) and so for some basic open \( X_{i(b)} \in FI(A)_\sigma \) we have \( y_{i(b)} \in X_{i(b)} \) and
\[ X_{i(b)} \subseteq b^{-1}(\{ c_{i(b)} \}). \]

24
Suppose that for each basic type \( \tau \) we establish the result using the induction hypothesis and one of the homeomorphisms.

(i) For some basic type \( \tau \), \( \forall a, b : \tau \) \( eval_A(a, b) \neq eval_A(b, a) \).

(ii) For some type \( A \), \( \forall a, b : A \) \( O_{\{a\}, A} \neq O_{\{b\}, A} \).

(iii) By the induction hypothesis, \( FI(A)_{A \to \tau} \) has a countable basis \( B_{\tau} \), and by assumption \( A_{\tau} \) is countable. So there are at most countably many subbasic open sets \( O_{U, a} \in FI(A)_{A \to \tau} \) for \( U \in B_{\tau} \) and \( a : A_{\tau} \). Thus \( FI(A)_{A \to \tau} \) has a countable basis.

Subinduction Step. Suppose that \( \tau \) is a product type or a function type. Then in each case we establish the result using the induction hypothesis and one of the homeomorphisms \( fp_A^{-1}, uc_A, gfp_A^{-1} \) and \( gcu_A^{-1} \).

3.17. Corollary. Suppose that \( \tau \) is a product type or a function type. Then in each case we establish the result using the induction hypothesis and one of the homeomorphisms \( fp_A^{-1}, uc_A, gfp_A^{-1} \) and \( gcu_A^{-1} \).

3.18. Lemma. Suppose that \( A \) is not extensional, i.e. \( A \not\models \text{Ext} \).

(i) For some basic type \( \tau \) and some type \( \sigma \) \( \exists a, b : A_{\sigma \to \tau} \) such that \( a \neq b \) but for all \( a_0 \in A_{\sigma} \), \( eval_A(a, a_0) = eval_A(b, a_0) \).

(ii) For some type \( \tau \) \( \in S \), the topology \( FI(A)_{\tau} \) is not a \( T_0 \) space.

(iii) For some basic type \( \tau \) \( \in B \) and some type \( \sigma \) \( \in S \) there exists \( a : A_{\sigma \to \tau} \) such that \( \{ a \} \) is not closed in \( FI(A)_{\sigma \to \tau} \).
Proof. (i) It suffices to show that for any types \( \sigma, \tau \in S \), if there exist \( a, b \in A_{(\sigma \to \tau)} \) such that \( a \neq b \), but for all \( a_0 \in A_\sigma \),
\[
eval_A(a, a_0) = \neval_A(b, a_0),
\]
then there exists \( \sigma' \in S \) and basic \( \tau' \in B \) and \( a', b' \in A_{(\sigma \to \tau)} \) such that \( a' \neq b' \) but for all \( a_0 \in A_\sigma \),
\[
eval_A(a', a_0) = \neval_A(b', a_0).
\]
This can be proved by induction on the complexity of \( \tau \).
(ii) By (i), for some basic type \( \tau \in B \) and \( \sigma \in S \) there exist \( a, b \in A_{(\sigma \to \tau)} \) such that \( a \neq b \) but for all \( a_0 \in A_\sigma \),
\[
eval_A(a, a_0) = \neval_A(b, a_0).
\]
So for any basic open set \( U \in FI(A)_\sigma \) and \( a_0 \in A_\tau \),
\[
a \in O_{U, a_0} \iff b \in O_{U, a_0}.
\]
Since \( a \neq b \) then \( FI(A)_{(\sigma \to \tau)} \) is not a \( T_0 \) space.
(iii) By (i), for some basic type \( \tau \in B \) and \( \sigma \in S \) there exist \( a, b \in A_{(\sigma \to \tau)} \) such that \( a \neq b \) but for all \( a_0 \in A_\sigma \),
\[
eval_A(a, a_0) = \neval_A(b, a_0).
\]
Thus \( b \) is adherent to \( \{ a \} \) but \( b \not\in \{ a \} \). So \( \{ a \} \) is not closed. \( \Box \)

4 Conservativity, Normal Forms and Term-Rewriting.

In this section we establish the main result of this paper, which gives a necessary and sufficient condition for infinitary higher-order equational logic to be conservative over first-order equational logic for ground first-order equations. We then apply this result to characterise the conservativity of finitary higher-order equational logic over first-order equational logic for arbitrary first-order equations. For these results we introduce a notion of observational equivalence on the elements of a higher-order algebra \( A \). Two elements \( a, b \in A_\tau \) of type \( \tau \in S \) are observationally equivalent when they cannot be distinguished by any basic open set \( U \), i.e. \( a \in U \iff b \in U \). Recall that basic open sets in the finite information topology are sets of elements containing the same finite information. Thus this notion of equivalence can with some justification be claimed to be observable. We will prove that conservativity arises precisely when observational equivalence forms a congruence on the initial model \( I(\Sigma, E) \) of a higher-order equational theory \( E \subseteq Eqn(\Sigma, X) \).

We begin by showing how the structure of infinitary higher-order equational proofs can be simplified, both by eliminating use of the projection rules, and by limiting the use of \( \omega \)-extensionality rules to certain structurally simple types. Then we show how the structure of such proofs can be further simplified by translating them into term rewriting proofs with \( \omega \)-rewrite steps, which generalise first-order term rewriting proofs to the higher-order case. Finally we prove our main results.

Our first lemma establishes that in the presence of the pairing operation and its axioms, the projection rule for product types can be eliminated from higher-order equational proofs.

**4.1. Lemma.** Let \( E \subseteq Eqn(\Sigma, X) \) be any higher-order equational theory containing the homeomorphism axioms \( \text{Hom} \). For any type \( \tau \in S \) and any terms \( t, t' \in T(\Sigma, X)_\tau \), if
\[
E \vdash_\omega t = t'
\]

26
then there exists a proof \( P \) of \( t = t' \) from \( E \) using the rules of infinitary higher-order equational logic without the projection rules 2.5.(v).

**Proof.** Consider any such \( E \subseteq Eqn(\Sigma, X) \) containing \( \text{Hom} \). We prove, by induction on the complexity of proofs, that for any proof \( P \) of an equation \( t = t' \) from \( E \) using the rules of infinitary higher-order equational logic there exists a proof \( P' \) of \( t = t' \) from \( E \) using the rules of infinitary higher-order equational logic without the projection rules.

**Basis.** (i) Consider any proof \( P \) consisting of a single axiom or a single use of the reflexivity rule. Then the result holds trivially.

**Induction Step.** (ii) Consider any proof \( P \) of \( t = t' \) from \( E \) in which the final step involves one of the rules of symmetry, transitivity, substitution or \( \omega \)-extensionality. Then the result follows trivially from the induction hypothesis.

(iii) Consider any proof \( P \) of \( t = t' \) from \( E \) in which the final step involves a projection rule. Then \( P \) has the form

\[
P_1 \quad P_2 \quad t = t'
\]

where \( P_1 \) is a proof of \( \text{proj}^1(t) = \text{proj}^1(t') \) and \( P_2 \) is a proof of \( \text{proj}^2(t) = \text{proj}^2(t') \). By the induction hypothesis there exist proofs \( P'_1 \) and \( P'_2 \) of \( \text{proj}^1(t) = \text{proj}^1(t') \) and \( \text{proj}^2(t) = \text{proj}^2(t') \) which do not make use of the projection rules. Thus using \( P'_1 \) and \( P'_2 \) and substitution we can derive

\[
\langle \text{proj}^1(t), \text{proj}^2(t) \rangle = \langle \text{proj}^1(t'), \text{proj}^2(t') \rangle,
\]

and hence using equation 3.3.(9.a), reflexivity and substitution, we can derive,

\[
t = t',
\]

without the use of any projection rules. \( \Box \)

Next we show that in the presence of currying, uncurrying, function-pairing and inverse function-pairing and their axioms, all instances of the \( \omega \)-extensionality rule in a proof can be replaced by instances of basic \( \omega \)-extensionality rules. (Recall that the \( \omega \)-extensionality rule for a type \( (\sigma \rightarrow \tau) \) is basic if, and only if, \( \tau \) is a basic type, \( \tau \in B \).)

**4.2. Lemma.** Let \( E \subseteq Eqn(\Sigma, X) \) be any higher-order equational theory containing the homomorphism axioms \( \text{Hom} \). For any type \( \tau \in S \) and any terms \( t, t' \in T(\Sigma, X)_{\tau} \), if

\[
E \vdash_{\omega} t = t'
\]

then there exists a proof \( P \) of \( t = t' \) from \( E \) using the rules of infinitary higher-order equational logic with instances of basic \( \omega \)-extensionality rules only.

**Proof.** We prove the result in two stages. Firstly we establish that for any higher-order equational theory \( E \subseteq Eqn(\Sigma, X) \) containing \( \text{Hom} \), and for any types \( \sigma, \tau \in S \) and any terms \( t, t' \in T(\Sigma, X)_{(\sigma \rightarrow \tau)} \), if there exists a proof \( P \) of \( t = t' \) from \( E \) in which the final inference uses the \( \omega \)-extensionality rule and any other instances of this rule are basic, then there exists a proof \( P' \) of \( t = t' \) in which all instances of the \( \omega \)-extensionality rule are basic.

We prove this result by induction on the complexity of \( \tau \).

**Basis.** (i) Suppose that \( \tau \in B \) is a basic type, then the result holds trivially.

**Induction Step.** (ii) Suppose that \( \tau \) is a product type, \( \tau = (\tau_1 \times \tau_2) \). Let \( P \) be a proof of \( t = t' \) from \( E \) in which the final step uses the \( \omega \)-extensionality rule and all other uses are basic. Then \( P \) has the form

\[
\frac{\langle P(t_0) \mid t_0 \in T(\Sigma)_{\sigma} \rangle}{t = t'}
\]

27
where for each \( t_0 \in T(\Sigma)_\sigma \), \( P(t_0) \) is a proof of

\[
\text{eval}(t, t_0) = \text{eval}(t', t_0)
\]

using basic \( \omega \)-extensionality rules only.

Consider any \( t_0 \in T(\Sigma)_\sigma \). Then, using \( P(t_0) \) and equations 3.3.(3.c), (9.b) and (9.c), we can derive

\[
\text{eval}(\text{proj}^1(fp^{-1}(t)), t_0) = \text{eval}(\text{proj}^1(fp^{-1}(t')), t_0)
\]

and

\[
\text{eval}(\text{proj}^2(fp^{-1}(t)), t_0) = \text{eval}(\text{proj}^2(fp^{-1}(t')), t_0).
\]

Since \( t_0 \) was arbitrarily chosen then we obtain proofs of

\[
\text{proj}^1(fp^{-1}(t)) = \text{proj}^1(fp^{-1}(t')),
\]

\[
\text{proj}^2(fp^{-1}(t)) = \text{proj}^2(fp^{-1}(t'))
\]

in which the final inference uses the \( \omega \)-extensionality rule for the types \((\sigma \rightarrow \tau_1)\) and \((\sigma \rightarrow \tau_2)\) respectively. So by the induction hypothesis there exist proofs \( P_1 \) and \( P_2 \) of these equations using basic \( \omega \)-extensionality rules only. Using such \( P_1 \) and \( P_2 \) and substitution we can derive

\[
\langle \text{proj}^1(fp^{-1}(t)), \text{proj}^2(fp^{-1}(t)) \rangle = \langle \text{proj}^1(fp^{-1}(t')), \text{proj}^2(fp^{-1}(t')) \rangle,
\]

and then, using equation 3.3.(9.a),

\[
fp^{-1}(t) = fp^{-1}(t'),
\]

and by substitution

\[
fp(fp^{-1}(t)) = fp(fp^{-1}(t')).
\]

Finally, using equation 3.3.(3.a), we can derive the equation \( t = t' \).

(iii) Suppose that \( \tau \) is a function type \( \tau = (\tau_1 \rightarrow \tau_2) \). Let \( P \) be a proof of \( t = t' \) from \( E \) in which the final inference uses the \( \omega \)-extensionality rule and all other instances are basic. Then \( P \) has the form

\[
\frac{(P(t_0) \mid t_0 \in T(\Sigma)_\sigma)}{t = t'}
\]

where for each \( t_0 \in T(\Sigma)_\sigma \), \( P(t_0) \) is a proof of

\[
\text{eval}(t, t_0) = \text{eval}(t', t_0)
\]

using basic \( \omega \)-extensionality rules only.

Consider any term \( t_0 \in T(\Sigma)_{(\sigma \times \tau_1)} \). Then using the proof \( P(\text{proj}^1(t_0)) \) we can derive

\[
\text{eval}(t, \text{proj}^1(t_0)) = \text{eval}(t', \text{proj}^1(t_0)),
\]

and hence

\[
\text{eval}(\text{eval}(t, \text{proj}^1(t_0)), \text{proj}^2(t_0)) = \text{eval}(\text{eval}(t', \text{proj}^1(t_0)), \text{proj}^2(t_0)).
\]

Then using equation 3.3.(1.c), we can derive

\[
\text{eval}(\text{uc}(t), \langle \text{proj}^1(t_0), \text{proj}^2(t_0) \rangle) = \text{eval}(\text{uc}(t'), \langle \text{proj}^1(t_0), \text{proj}^2(t_0) \rangle)
\]

and then using equation 3.3.(9.a),

\[
\text{eval}(\text{uc}(t), t_0) = \text{eval}(\text{uc}(t'), t_0).
\]
Since $t_0$ was arbitrarily chosen we can derive

$$uc(t) = uc(t')$$

from $E$, where the final inference uses the $\omega$-extensionality rule for the type $((\sigma \times \tau_1) \to \tau_2)$ and all other instances of this rule are basic. Thus by the induction hypothesis there exists a proof $P'$ of $uc(t) = uc(t')$ using basic $\omega$-extensionality rules only. Using such $P'$ and substitution we can derive

$$cu(uc(t)) = cu(uc(t'))$$

and hence by equation 3.3.(1.a),

$$t = t'.$$

This completes the induction.

We now prove the main lemma using the above fact, by induction on the complexity of proofs.

**Basis.** (i) Consider any proof $P$ of $t = t'$ from $E$ consisting of a single axiom or a single use of the reflexivity rule. Then the result holds trivially.

**Induction Step.** (ii) Consider any proof $P$ of $t = t'$ from $E$ in which the final inference uses one of the rules of symmetry, transitivity, substitution or projection. Then the result follows trivially from the induction hypothesis.

(iii) Consider any proof $P$ of $t = t'$ from $E$ in which the final inference uses the $\omega$-extensionality rule for a type $(\sigma \to \tau)$. Then $P$ has the form

$$\langle P(t_0) \mid t_0 \in T(\Sigma, \sigma) \rangle$$

where for each $t_0 \in T(\Sigma, \sigma)$, $P(t_0)$ is a proof of

$$eval(t, t_0) = eval(t', t_0).$$

By the induction hypothesis, for each $t_0 \in T(\Sigma, \sigma)$ there exists a proof $P'(t_0)$ of

$$eval(t, t_0) = eval(t', t_0)$$

from $E$ using basic $\omega$-extensionality rules only. So

$$\langle P'(t_0) \mid t_0 \in T(\Sigma, \sigma) \rangle$$

is a proof of $t = t'$ in which the final inference uses the $\omega$-extensionality rule and all other instances of this rule are basic. Thus by the above result there exists a proof $Q$ of $t = t'$ in which all instances of the $\omega$-extensionality rule are basic.

To simplify the structure of infinitary higher-order equational proofs even further, we introduce the notion of a *term rewriting proof with $\omega$-rewrite steps*. This proof system extends the correspondence between many-sorted first-order equational logic and many-sorted term rewriting to the infinitary higher-order case in an obvious way. However, the structure of a term rewriting proof imposes a normal form on the use of the transitivity rule that is useful for proving our main theorem.

We assume the reader is familiar with the definitions of an occurrence $\bar{i} \in \mathbb{N}^*$ in a term $t \in T(\Sigma, X)$ and the set $Occ(t)$ of all occurrences in $t$, the *subterm of $t$ at the occurrence* $\bar{i} \in Occ(t)$, denoted by $t(\bar{i})$, and the *substitution* of a term $t'$ at $\bar{i}$ in $t$, denoted by $t(\bar{i}/t')$. The reader may consult any introductory text on term rewriting, for example Le Chenadec [1986].
4.3. Definition. Let $E \subseteq \text{Eqn}(\Sigma, X)$ be any higher-order equational theory.

(i) A rewrite step $s$ from $E$ is a five tuple

$$s = (t, \bar{i}, t_t = t_r, \alpha, t'),$$

where for some types $\sigma, \tau \in S$, terms $t, t' \in T(\Sigma, X)_{\tau}$, occurrence $\bar{i} \in \text{Occ}(t)_{\sigma}$, terms $t_t, t_r \in T(\Sigma, X)_{\sigma}$ and assignment $\alpha : X \rightarrow T(\Sigma, X)$, either: (a) $t_t$ is $t_r$, or (b) $t_t = t_r \in E$, or (c) $t_r = t_t \in E$, and in each case $t(\bar{i}) = \overline{\alpha}(t_t)$ and $t' = t(\bar{i}/\overline{\alpha}(t_r))$. We define the degree of $s$ to be $\deg(s) = 0$.

(ii) An $\omega$-rewrite step $s$ from $E$ is a six tuple

$$s = (t, \bar{i}, t_t = t_r, \bar{s}, \alpha, t'),$$

where for some type $\tau \in S$, $t, t' \in T(\Sigma, X)_{\tau}$ are terms and for some function type $(\sigma \rightarrow \delta) \in S$ and occurrence $\bar{j} \in \text{Occ}(t)(\sigma\rightarrow\delta)$, and terms $t_t, t_r \in T(\Sigma, X)_{(\sigma\rightarrow\delta)}$

$$\bar{s} = (\overline{s}(t_0) \mid t_0 \in T(\Sigma)_{\sigma})$$

is a family of rewrite proofs, $\overline{s}(t_0)$ being a rewrite proof of $\text{eval}(t_t, t_0) = \text{eval}(t_r, t_0)$ from $E$ with $\omega$-rewrite steps for each $t_0 \in T(\Sigma)_{\sigma}$, and $\alpha : X \rightarrow T(\Sigma, X)$ is an assignment such that $t(\bar{i}) = \overline{s}(t_t)$ and $t' = t(\bar{i}/\overline{s}(t_r))$. We define the degree of $s$ to be

$$\deg(s) = 1 + \sup\{ \deg(\overline{s}(t_0)) \mid t_0 \in T(\Sigma)_{\sigma} \}.$$

We say that $s$ is a basic $\omega$-rewrite step if, and only if, $\delta \in B$ is a basic type.

(iii) A rewrite proof $P$ from $E$ with $\omega$-rewrite steps is a non-empty finite sequence

$$P = s_1, \ldots, s_n,$$

where for each $1 \leq j \leq n$, either:

(a) $s_j = (t_j, \bar{i}_j, t'_j = t'_j, \alpha_j, t'_j)$, is a rewrite step from $E$, or

(b) $s_j = (t_j, \bar{i}_j, t'_j = t'_j, \bar{s}_j, \alpha_j, t'_j)$ is an $\omega$-rewrite step from $E$,

and for each $1 \leq j \leq n - 1$, $t'_j$ is $t_{j+1}$. We say that $P$ is a rewrite proof of $t_1 = t'_n$ from $E$ with $\omega$-rewrite steps. We define the degree of $P$ to be

$$\deg(P) = \sup\{ \deg(s_1), \ldots, \deg(s_n) \}.$$

If there exists a rewrite proof $P$ of $t = t'$ from $E$ with $\omega$-rewrite steps of degree $\alpha \in \text{Ord}$ then we write

$$t \overset{E, \omega, \alpha}{\rightarrow} t',$$

or simply, if the order of the rewrite proof is irrelevant,

$$t \overset{E, \omega}{\rightarrow} t'.$$

If $P$ has degree 0 then we may write

$$t \overset{E}{\rightarrow} t',$$

following the conventional notation, and say that $P$ is a rewrite proof from $E$ of $t = t'$.

The well known correspondence between term rewriting proofs and derivations in first-order equational logic extends to rewrite proofs with $\omega$-rewrite steps and infinitary higher-order equational logic in the obvious way.
4.4. **Correspondence Theorem.** Let $E \subseteq \text{Eqn}(\Sigma, X)$ be any higher-order equational theory. For any type $\tau \in S$ and any terms $t, t' \in T(\Sigma, X)_\tau$:

(i) $E \vdash t = t' \iff t \xrightarrow{E} t'$,

(ii) for any $\alpha \in \text{Ord}$, $E \vdash \omega, \alpha t = t' \iff t \xrightarrow{E, \omega, \alpha} t'$,

(iii) $t = t'$ is provable from $E$ using the infinitary rules of higher-order equational logic and basic $\omega$-extensionality rules only if, and only if, there exists a rewrite proof of $t = t'$ from $E$ using basic $\omega$-rewrite steps only.

**Proof.** (i) See for example Ehrig and Mahr [1985]. (ii) By induction on the degree of infinitary proofs and rewrite proofs (using (i) as the induction basis), and Lemma 4.1. (iii) By inspection of the proof of (ii). □

We will require an elementary fact about rewrite proofs (without $\omega$-rewrite steps) of ground equations.

4.5. **Proposition.** Let $E \subseteq \text{Eqn}(\Sigma, X)$ be any higher-order equational theory. For any type $\tau \in S$ and any ground terms $t, t' \in T(\Sigma)_\tau$, if there exists a rewrite proof of $t = t'$ of length $n$ from $E$ (without $\omega$-rewrite steps) then there exists a rewrite proof of $t = t'$ of length $n$ where all intermediary terms are ground.

**Proof.** By induction on $n$. □

Next we introduce our notion of observational equivalence on elements of a higher-order algebra.

4.6. **Definition.** Let $A$ be any $\Sigma$ algebra (not necessarily extensional). For each $\tau \in S$, we define the relation $\equiv_{\tau}^{\text{obs}} \subseteq A^2_\tau$ of **observational equivalence** by,

$$a \equiv_{\tau}^{\text{obs}} b \iff \text{for all subbasic open } U \in \text{FI}(A)_\tau, a \in U \iff b \in U.$$  

Clearly observational equivalence is an equivalence relation on elements. By definition $a, b \in A_\tau$ are observationally equivalent if, and only if, no open set, in particular no basic open set (i.e. finite observation) can separate them. Note that since the subbasic open sets of the finite information topology can be defined for any algebra $A$ of signature $\Sigma$ (irrespective of whether $A$ is extensional or not) then the notion of observational equivalence can also be defined for any $\Sigma$ algebra $A$. In particular, observational equivalence can always be defined on elements of the initial algebra $I(\Sigma, E)$ of an equational theory $E \subseteq \text{Eqn}(\Sigma, X)$ (which is not normally extensional).

First we note that observational equivalence is a finer equivalence relation on terms than provable equivalence using infinitary higher-order equational logic.

4.7. **Lemma.** Let $E \subseteq \text{Eqn}(\Sigma, X)$ be any equational theory. For any $\tau \in S$, and any ground terms $t, t' \in T(\Sigma)_\tau$,

$$t \equiv_{\tau}^{\text{obs}} t' \Rightarrow E \vdash \omega t = t'.$$

**Proof.** By induction on the complexity of $\tau$. □

Now we can establish our main result concerning conservativity.

4.8. **Conservativity Theorem.** Let $E \subseteq \text{Eqn}(\Sigma, X)$ be any higher-order equational theory which contains the homeomorphism axioms $\text{Hom}$. Then infinitary higher-order equational logic...
is conservative over equational logic on $E$ for ground first-order equations if, and only if, observational equivalence $\equiv^{obs}$ is a congruence on the (first-order) initial model $I(\Sigma,E)$.

**Proof.** $\Rightarrow$ We prove the contrapositive by showing that for any type $\tau \in S$ and any term $t(x_1, \ldots, x_n) \in T(\Sigma, X)_{\tau}$ with $n$ free variables $x_i \in X_{\tau(i)}$ for $1 \leq i \leq n$, and for any ground terms $t_i, t_i' \in T(\Sigma)_{\tau(i)}$ for $1 \leq i \leq n$, if

$$[t_i] \equiv^{obs}_{r(i)} [t_i']$$

for each $1 \leq i \leq n$, but

$$[t(t_1, \ldots, t_n)] \not\equiv^{obs}_{\tau} [t(t_1', \ldots, t_n')]$$

then there exists a basic type $\tau' \in B$ and ground terms $t_0, t_0' \in T(\Sigma)_{\tau}$ such that

$$E \vdash \omega t_0 = t_0'$$

but

$$E \nvdash t_0 = t_0'.$$

We prove this by induction on the complexity of $\tau$.

**Basis.** (i) Suppose that $\tau \in B$ is a basic type. By assumption $[t_i] \equiv^{obs}_{r(i)} [t_i']$ for $1 \leq i \leq n$, so by Lemma 4.7, $E \vdash \omega t_i = t_i'$ for each $1 \leq i \leq n$. So by substitution,

$$E \vdash \omega t(t_1, \ldots, t_n) = t(t_1', \ldots, t_n').$$

But also by assumption

$$[t(t_1, \ldots, t_n)] \not\equiv^{obs}_{\tau} [t(t_1', \ldots, t_n')]$$

so

$$E \nvdash t(t_1, \ldots, t_n) = t(t_1', \ldots, t_n')$$

since $\tau$ is basic. Thus taking $\tau' = \tau$, $t_0 = t(t_1, \ldots, t_n)$ and $t_0' = t(t_1', \ldots, t_n')$ then the result follows.

**Induction Step.** (ii) Consider any product type $\tau = (\sigma \times \delta)$ and suppose

$$[t(t_1, \ldots, t_n)] \not\equiv^{obs}_{(\sigma \times \delta)} [t(t_1', \ldots, t_n')]$$

but $[t_i] \equiv^{obs}_{r(i)} [t_i']$ for $1 \leq i \leq n$. Then there exists a subbasic open set $U \in FI(I(\Sigma,E))_{(\sigma \times \delta)}$ such that either

$$[t(t_1, \ldots, t_n)] \in U \text{ and } [t(t_1', \ldots, t_n)] \notin U \quad (1),$$

or

$$[t(t_1, \ldots, t_n)] \notin U \text{ and } [t(t_1', \ldots, t_n)] \in U \quad (2).$$

Suppose that $(1)$ holds and that $U$ has the form $\langle V_1, C(I(\Sigma,E))_{\delta} \rangle_{I(\Sigma,E)}$ for some open $V_1 \in FI(I(\Sigma,E))_{\sigma}$. Then

$$[\text{proj}^1(t(t_1, \ldots, t_n))] \in V_1 \text{ and } [\text{proj}^1(t(t_1', \ldots, t_n'))] \notin V_1.$$ 

Thus

$$[\text{proj}^1(t(t_1, \ldots, t_n))] \not\equiv^{obs}_{\tau'} [\text{proj}^1(t(t_1', \ldots, t_n'))].$$

Thus by the induction hypothesis there exists $\tau' \in B$ and $t_0, t_0' \in T(\Sigma)_{\tau'}$ such that

$$E \vdash \omega t_0 = t_0'.$$
but $E \not\vdash t_0 = t'_0$. Similarly the result holds if $U$ has the form $\langle C(I(\Sigma, E))_\delta, V_2 \rangle_I(\Sigma, E)$ for some open $V_2 \in FI(I(\Sigma, E))_\delta$. If (2) holds then the result follows by symmetry of $t_i$ and $t'_i$ for $1 \leq i \leq n$.

(iii) Consider any function type $\tau = (\sigma \rightarrow \delta)$. We prove the result by subinduction on the complexity of $\delta$.

**Subbasis.** (iii.a) Suppose $\delta \in B$ is a basic type and $[t_i] \equiv_{\tau(i)}^{obs} [t'_i]$ for $1 \leq i \leq n$ but

$$t(t_1, \ldots, t_n) \not\equiv_{(\sigma \rightarrow \delta)}^{obs} t(t'_1, \ldots, t'_n).$$

Then there exists a subbasic open set $U \in FI(I(\Sigma, E))_{(\sigma \rightarrow \delta)}$ with either

$$[t(t_1, \ldots, t_n)] \in U \text{ and } [t(t'_1, \ldots, t'_n)] \not\in U \quad (1),$$

or

$$[t(t_1, \ldots, t_n)] \not\in U \text{ and } [t(t'_1, \ldots, t'_n)] \in U \quad (2).$$

Suppose that (1) holds. Then for some non-empty open set $V \in FI(I(\Sigma, E))_\sigma$ and element $a \in I(\Sigma, E)_\delta$ we have

$$[t(t_1, \ldots, t_n)] \in O_V \{a\} \text{ and } [t(t'_1, \ldots, t'_n)] \not\in O_V \{a\}.$$

So for some $t' \in T(\Sigma)_\sigma$ with $[t'] \in V$ we have

$$eval_{I(\Sigma, E)}([t(t_1, \ldots, t_n)], [t']) = a \text{ and } eval_{I(\Sigma, E)}([t(t'_1, \ldots, t'_n)], [t']) \not= a.$$

So

$$[eval(t(t_1, \ldots, t_n), t')] \not= [eval(t(t'_1, \ldots, t'_n), t')]$$

and hence

$$[eval(t(t_1, \ldots, t_n), t')] \not= \delta^{obs} [eval(t(t'_1, \ldots, t'_n), t')]$$

since $\delta$ is basic. By assumption $[t_i] \equiv_{\tau(i)}^{obs} [t'_i]$ for $1 \leq i \leq n$, so the result follows from the induction basis. Similarly if (2) holds then the result follows from the symmetry of $t_i$ and $t'_i$ for $1 \leq i \leq n$.

**Subinduction Step.** (iii.b) Suppose that $\delta = (\delta_1 \times \delta_2)$ is a product type, and

$$[t_i] \equiv_{(\sigma \rightarrow (\delta_1 \times \delta_2))}^{obs} [t'_i]$$

for $1 \leq i \leq n$, but

$$[t(t_1, \ldots, t_n)] \not= [t(t', \ldots, t'_n)].$$

Then there exists a subbasic open set $U \in FI(I(\Sigma, E))_{(\sigma \rightarrow (\delta_1 \times \delta_2))}$ with either

$$[t(t_1, \ldots, t_n)] \in U \text{ and } [t(t'_1, \ldots, t'_n)] \not\in U \quad (1),$$

or

$$[t(t_1, \ldots, t_n)] \not\in U \text{ and } [t(t'_1, \ldots, t'_n)] \in U \quad (2).$$

Suppose that (1) holds, then

$$[fp^{-1}(t(t_1, \ldots, t_n))] \in fp^{-1}_{I(\Sigma, E)}(U) \text{ and } [fp^{-1}(t(t'_1, \ldots, t'_n))] \not\in fp^{-1}_{I(\Sigma, E)}(U).$$

Thus

$$[fp^{-1}(t(t_1, \ldots, t_n))] \not= [fp^{-1}(t(t'_1, \ldots, t'_n))].$$

So by the subinduction hypothesis there exists $\tau' \in B$ and $t_0$, $t'_0 \in T(\Sigma)_{\tau'}$ such that

$$E \vdash \omega \ t_0 = t'_0$$
but
\[ E \not\vdash t_0 = t'_0. \]

Similarly the result follows if (2) holds by symmetry of \( t_i \) and \( t'_i \) for \( 1 \leq i \leq n \).

(iii.c) Suppose that \( \delta = (\delta_1 \rightarrow \delta_2) \) is a function type. We prove the result by a subinduction on the complexity of \( \delta_2 \). the proof in each of the three cases is similar to (iii.b) above using the appropriate operator \( (uc_{I(\Sigma, E)}, gf_{I(\Sigma, E)}^{-1}, \text{and } gcu_{I(\Sigma, E)}^{-1}) \) respectively and is omitted.

\[ \iff \] We prove the contrapositive, i.e. we show that if infinitary higher-order equational logic is not conservative over equational logic on \( E \) for ground first-order equations then \( \equiv^{obs} \) is not a congruence on \( I(\Sigma, E) \).

Suppose for some basic type \( \tau \in \mathcal{B} \) and ground first-order terms \( t, t' \in T(\Sigma)_{\tau} \) that
\[ E \vdash \omega \ t = t' \]
but
\[ E \not\vdash t = t'. \]

Then by Lemma 4.2 and Correspondence Theorem 4.4.(iii) there exists a rewrite proof of \( t = t' \) from \( E \) using basic \( \omega \)-rewrite steps only,
\[ t \xrightarrow{E; \omega, \beta} t' \]
for some \( \beta \in \text{Ord} \) with \( \beta > 0 \), but
\[ t \xleftarrow{E} t'. \]

We prove the result by transfinite induction on the degree of rewrite proofs using basic \( \omega \)-rewrite steps only.

**Basis.** Suppose that \( \beta = 1 \) and that
\[ t \xrightarrow{E; \omega, 1} t' \]
but
\[ t \xleftarrow{E} t'. \]

Let \( \overline{s} = s_1, \ldots, s_k \) be a rewrite proof of \( t = t' \) with basic \( \omega \)-rewrite steps of degree 1. Then for some \( 1 \leq j \leq k \), \( s_j \) is a basic \( \omega \)-rewrite step of degree 1,
\[ s_j = (t_j, \overline{t}_j, t'_j, \overline{t}'_j, \alpha_j, t'_j), \]
where for some function type \( (\sigma \rightarrow \delta) \in \mathcal{S} \) with \( \delta \in \mathcal{B} \), we have \( t_j, t'_j, t_0 \in T(\Sigma, X)_{(\sigma \rightarrow \delta)} \) and
\[ \overline{s_j} = \langle \overline{s_j}(t_0) \mid t_0 \in T(\Sigma)_\sigma \rangle \]
is a family of rewrite proofs such that for each \( t_0 \in T(\Sigma)_\sigma \),
\[ \overline{s_j}(t_0) = \overline{s_j}(t_0)_1, \ldots, \overline{s_j}(t_0)_{k_j(t_0)} \]
is a rewrite proof of length \( k_j(t_0) \) of degree 0 (i.e. without \( \omega \)-rewrite steps) of the equation
\[ \text{eval}(t'_j, t_0) = \text{eval}(t'_j, t_0). \]

Furthermore since \( t \xleftarrow{E} t' \) it follows that
\[ t_j \xrightarrow{E} t'_j \] (1).
By Proposition 4.5, \( S \) can be chosen so that \( t_j \) and \( t'_j \) are ground. Therefore \( \overline{\sigma_j}(t'_j) \) and \( \overline{\tau_j}(t'_j) \) are ground. Now by the Correspondence Theorem 4.4.(i), for each ground term \( t_0 \in T(\Sigma)_{\sigma} \),

\[
E \vdash eval(t'_j, t_0) = eval(t'_j, t_0).
\]

So for each \( t_0 \in T(\Sigma)_{\sigma} \) by substitution,

\[
E \vdash eval(\overline{\sigma_j}(t'_j), t_0) = eval(\overline{\tau_j}(t'_j), t_0)
\] (2).

Suppose there is no subbasic open \( U \in FI(I(\Sigma, E))(\sigma \rightarrow \delta) \) with \( [\overline{\tau_j}(t'_j)] \in U \). Then clearly \( [\overline{\sigma_j}(t'_j)] \) is discontinuous. Hence \( I(\Sigma, E) \) has discontinuous carrier sets.

Suppose there is a subbasic open set \( U \in FI(I(\Sigma, E))(\sigma \rightarrow \delta) \) with \( [\overline{\tau_j}(t'_j)] \in U \). Consider any such set \( U \). Then by definition for some open set \( V \in FI(I(\Sigma, E))_{\sigma} \) and \( t' \in T(\Sigma)_{\delta} \), \( U = O_{V, \{t'\}} \).

Consider any \( \{t_0\} \in V \). Then by definition,

\[
eval_{I(\Sigma, E)}([\overline{\tau_j}(t'_j)], \{t_0\}) = \{t'\}.
\]

So \( E \vdash eval(\overline{\tau_j}(t'_j), t_0) = \{t'\} \). Thus by (2), \( E \vdash eval(\overline{\sigma_j}(t'_j), t_0) = \{t'\} \). So

\[
eval_{I(\Sigma, E)}([\overline{\sigma_j}(t'_j)], \{t_0\}) = \{t'\}.
\]

Since \( \{t_0\} \) was arbitrarily chosen then \( [\overline{\tau_j}(t'_j)] \) \( \in U \). Since \( U \) was arbitrarily chosen, by symmetry of \( t'_j \) and \( t'_j \),

\[
[\overline{\sigma_j}(t'_j)] \equiv^{\text{obs}}_{(\sigma \rightarrow \delta)} [\overline{\tau_j}(t'_j)].
\]

Suppose for a contradiction that \( \equiv^{\text{obs}} \) is a congruence. Then

\[
[t_j] \equiv^{\text{obs}} [t_j(\overline{t_j}/\overline{\tau_j}(t'_j))]
\]

but since \( \tau \) is a basic type it follows that

\[
[t_j] = [t_j(\overline{t_j}/\overline{\sigma_j}(t'_j))]
\]

i.e.

\[
E \vdash t_j = t_j(\overline{t_j}/\overline{\tau_j}(t'_j)),
\]

which contradicts (1) above. Thus \( \equiv^{\text{obs}} \) is not a congruence.

**Induction Step.** Suppose that \( \beta > 1 \) and that

\[
t \xrightarrow{E, \omega, \beta} t'
\]

with only basic \( \omega \)-rewrite steps, but

\[
t \not\xrightarrow{E} t'.
\]

Let \( \pi = s_1, \ldots, s_k \) be a rewrite proof with basic \( \omega \)-steps of the equation \( t = t' \) with degree \( \beta \). Then for some \( 1 \leq j \leq k \), \( s_j \) is a basic \( \omega \)-rewrite step of degree \( \beta_j \leq \beta \),

\[
s_j = (t_j, \overline{t_j}, t'_j, \overline{\pi}, \alpha_j, t'_j),
\]

where for some function type \( (\sigma \rightarrow \delta) \in S \) with \( \delta \in B \) we have \( t'_j, t'_j \in T(\Sigma, X)_{(\sigma \rightarrow \delta)} \) and

\[
\overline{\pi_j} = \langle \overline{\pi_j}(t_0) \mid t_0 \in T(\Sigma)_{\sigma} \rangle
\]

35
is a family of rewrite proofs, such that for each \( t_0 \in T(\Sigma)_\sigma \),
\[
\overline{s}_j(t_0) = \overline{s}_j(t_0)_1, \ldots, \overline{s}_j(t_0)_{k_j(t_0)}
\]
is a rewrite proof with basic \( \omega \)-rewrite steps of the equation
\[
eval(t^j_1, t_0) = \overline{\eval}(t^j_1, t_0)
\]
of degree \( \beta_j(t_0) < \beta_j \). Also
\[
t_j \not\xrightarrow{E} t_j'
\]
Again by Proposition 4.5, \( \overline{S} \) can be chosen so that \( t_j \) and \( t'_j \) are ground. Therefore \( \overline{\eval}(t^j_1) \) and \( \overline{\eval}(t^j_1) \) are ground.

By the Correspondence Theorem 4.4.(iii), for each \( t_0 \in T(\Sigma)_\sigma \),
\[
E \vdash_{\omega, \beta_j(t_0)} \eval(t^j_1, t_0) = \eval(t^j_1, t_0)
\]
using only basic \( \omega \)-evaluation rules, and so by substitution, for each \( t_0 \in T(\Sigma)_\sigma \),
\[
E \vdash_{\omega, \beta_j(t_0)} \eval(\overline{\eval}(t^j_1), t_0) = \eval(\overline{\eval}(t^j_1), t_0).
\]
Thus by the Correspondence Theorem 4.4.(iii), for each \( t_0 \in T(\Sigma)_\sigma \),
\[
\eval(\overline{\eval}(t^j_1), t_0) \xrightarrow{E, \omega, \beta_j(t_0)} \eval(\overline{\eval}(t^j_1), t_0)
\]
using basic \( \omega \)-rewrite steps only.

Suppose that for some \( t_0 \in T(\Sigma)_\sigma \),
\[
\eval(\overline{\eval}(t^j_1), t_0) \not\xrightarrow{E} \eval(\overline{\eval}(t^j_1), t_0).
\]
Then by (4) and the induction hypothesis, \( \equiv^{\text{obs}} \) is not a congruence. Suppose that for each \( t_0 \in T(\Sigma)_\sigma \),
\[
\eval(\overline{\eval}(t^j_1), t_0) \not\xrightarrow{E} \eval(\overline{\eval}(t^j_1), t_0).
\]
Then again as for the induction basis we have \( \overline{\eval}(t^j_1) \equiv^{\text{obs}}(\sigma \rightarrow \delta) \overline{\eval}(t^j_1) \). So the assumption that \( \equiv^{\text{obs}} \) is a congruence contradicts (3).

Since our notion of observational equivalence is somewhat unfamiliar, it is natural to seek stronger conditions which imply the congruence property, and which themselves seem more natural or easier to verify. One such natural condition on a higher-order algebra is continuity, and this condition is indeed strong enough to imply the congruence property.

4.9. Lemma. For any \( \Sigma \) algebra \( A \), if \( A \) is continuous in the finite information topology then \( \equiv^{\text{obs}} \) is a congruence on \( A \).

Proof. Consider any \( n \geq 1 \), any \( w = \tau(1) \ldots \tau(n) \in S^+ \) and any function symbol \( f \in \Sigma_{w, \tau} \).
Consider any \( a_i, b_i \in A_{\tau(i)} \) for \( 1 \leq i \leq n \) and suppose
\[
a_i \equiv^{\text{obs}}_{\tau(i)} b_i,
\]
for \( 1 \leq i \leq n \). We must show that
\[
f_A(a_1, \ldots, a_n) \equiv^{\text{obs}}_{\tau} f_A(b_1, \ldots, b_n)
\]
(1).
Now consider any subbasic open set \( U \in FI(A) \), and suppose that
\[
f_A(a_1, \ldots, a_n) \in U.
\]
Since \( A \) is continuous in the finite information topology there exists \( V_i \in Nbd(a_i) \) for each \( 1 \leq i \leq n \) such that for all \( v_i \in V_i \) for \( 1 \leq i \leq n \),
\[
f_A(v_1, \ldots, v_n) \in U.
\]
Now since \( a_i \equiv_{\tau(i)}^\text{obs} b_i \) for each \( 1 \leq i \leq n \) then \( b_i \in V_i \) for each \( 1 \leq i \leq n \). Thus
\[
f_A(b_1, \ldots, b_n) \in U.
\]
Since \( U \) was arbitrarily chosen, then (1) holds by the symmetry of \( a_i \) and \( b_i \). \( \square \)

4.10. Corollary. Let \( E \subseteq Eqn(\Sigma, X) \) be any higher-order equational theory which contains the homeomorphism axioms \( \text{Hom} \). If \( I(\Sigma, E) \) is continuous in the finite information topology then infinitary higher-order equational logic is conservative on \( E \) for ground first-order equations.

Proof. Immediate from Theorem 4.8 and Lemma 4.9. \( \square \)

Next we turn to the characterisation of conservativity for finitary higher-order equational logic and the existence of normal form proofs. We begin by considering the well known folk theorem of first-order equational logic (sometimes called the Theorem on Constants) which states that variable symbols can be exchanged for (or simply reinterpreted as) fresh constant symbols in a signature without altering the provability of formulas. If \( \Sigma \) is any \( S \)-sorted signature and \( X = \langle X_s \mid s \in S \rangle \) is an \( S \)-indexed family of sets of variable symbols (with \( X_s \) disjoint from \( \Sigma \lambda, s \) for each \( s \in S \)) then we can define the \( S \)-sorted signature \( \Sigma \cup X \), where for each \( w \in S^* \) and \( s \in S \):
\[
(S \cup X)_{w, s} = \begin{cases} 
\Sigma\lambda, s \cup X_s, & \text{if } w = \lambda; \\
\Sigma_{w, s}, & \text{otherwise.}
\end{cases}
\]

Then every equation \( e \in Eqn(\Sigma, X) \) is also a ground equation over \( \Sigma \cup X \), and every equational theory \( E \subseteq Eqn(\Sigma, X) \) is also a ground equational theory over \( \Sigma \cup X \). To express the Theorem on Constants precisely, as well as its generalisation to the higher-order case, it is necessary to make explicit two further parameters of the inference relations. Thus, in the sequel, we let \( \vdash_{\Sigma, X} \) (respectively \( \vdash_{\Sigma, X}^{\text{eval}}, \vdash_{\Sigma, X}^{\omega} \)) denote provability in first-order many-sorted equational logic (respectively finitary higher-order equational logic, infinitary higher-order equational logic) with respect to the signature \( \Sigma \) and family \( X \) of sets of variables.

4.11. Theorem on Constants. Let \( Y = \langle Y_\tau \mid \tau \in S \rangle \) be any \( S \)-indexed family of infinite sets of variable symbols disjoint from \( X \) and \( \Sigma_\tau \) (i.e. \( Y_\tau \cap (\Sigma \cup X)_{\lambda, \tau} = \emptyset \) for each \( \tau \in S \)). Then for any higher-order equational theory \( E \subseteq Eqn(\Sigma, X) \) and any equation \( e \in Eqn(\Sigma, X) \):

(i) \( E \vdash_{\Sigma, X}^{\text{eval}} e \iff E \vdash_{\Sigma \cup X, Y}^{\text{eval}} e \),

(ii) \( E \vdash_{\Sigma, X}^{\omega} e \iff E \vdash_{\Sigma \cup X, Y}^{\omega} e \).

Proof. The proofs of (i) and (ii) are entirely similar. (i) Follows from Completeness Theorems 2.6 and 2.9. (ii) Follows from the Completeness Theorem for first-order many-sorted equational logic (see for example Meinke and Tucker [1992]). \( \square \)
The Theorem on Constants for higher-order equational logic (4.11.(i)) is quite different to the first-order result (4.11.(ii)), by virtue of the form of the extensionality and ω-extensionality rules. The free variable in the premise of the finitary extensionality rule, associated with implicit universal quantification, becomes a fresh constant symbol when variables are re-interpreted as constants. Thus the Theorem on Constants for higher-order equations turns out to relate finitary and infinitary higher-order equational logic to each other. In particular, it allows us to relate the conservativity properties of these two logics over first-order equational logic. Thus we obtain the following characterisation of conservativity for finitary higher-order equational logic, and simultaneously a characterisation of the existence of eval normal form proofs for this logic. (Recall Proposition 2.14.)

4.12. Normal Form Theorem. Let $E \subseteq Eqn(\Sigma, X)$ be any higher-order equational theory which contains the homeomorphism axioms $\text{Hom}$. The following are equivalent:

(i) finitary higher-order equational logic is conservative on $E$ for first-order equations;

(ii) for every equation $e \in Eqn(\Sigma, X)$ (of any order), if $E \vdash_{\text{eval}} e$ then there is a finitary proof $P$ of $e$ which is in eval normal form;

(iii) observational equivalence, $\equiv^{\text{obs}}$, is a congruence on the free algebra $T_E(\Sigma, X)$.

**Proof.** The equivalence of (i) and (ii) is simply Proposition 2.14. We need only consider the equivalence of (i) and (iii). Now by the Theorem on Constants, 4.11, finitary higher-order equational logic is conservative on first-order equations $e \in Eqn(\Sigma, X)$ if, and only if, infinitary higher-order equational logic is conservative on ground first-order equations $e \in Eqn(\Sigma \cup X, Y)$, where $Y = \{Y_\tau \mid \tau \in S\}$ is an $S$-indexed family of sets of variable symbols disjoint from $X$ and $\Sigma$, (i.e. $Y_\tau \cap (\Sigma_{\lambda, \tau} \cup X_\tau) = \emptyset$ for each $\tau \in S$). Recall that the free algebra $T_E(\Sigma, X)$ can be concretely constructed as the quotient term algebra $T(\Sigma, X)/\equiv_E$, where $\equiv_E$ is the congruence on terms induced by provable equivalence using first-order many-sorted equational logic. Noting that $\equiv^{\text{obs}}$ is a congruence on $T(\Sigma \cup X)/\equiv_E$ if and only if, $\equiv^{\text{obs}}$ is a congruence on $T(\Sigma, X)/\equiv_E$ then the result follows from the Conservativity Theorem 4.8.

4.13. Corollary. Let $E \subseteq Eqn(\Sigma, X)$ be any higher-order equational theory which contains the homeomorphism axioms $\text{Hom}$. If the free algebra $T_E(\Sigma, X)$ is continuous in the finite information topology then:

(i) finitary higher-order equational logic is conservative on $E$ for first-order equations, and,

(ii) for every equation $e \in Eqn(\Sigma, X)$ (of any order), if $E \vdash_{\text{eval}} e$ then there is a finitary proof $P$ of $e$ which is in eval normal form;

**Proof.** Immediate from Lemma 4.9 and Theorem 4.12.

5 References.

A. Church, A formulation of the simple theory of types, Journal of Symbolic Logic 5 (1940), 56-68.
K. Meinke, Universal algebra in higher types, Theoretical Computer Science 100 (1992), 385-417.