PARTIAL HIGHER-ORDER SPECIFICATIONS

Egidio Astesiano and Maura Cerioli
Dipartimento di Matematica - Università di Genova
Via L.B. Alberti 4 - 16132 Genova Italy
e-mail {Astes,Cerioli}@igecuniv.bitnet

Abstract. *In this paper we study the classes of extensional models of higher-order partial conditional specifications. After investigating the closure properties of these classes, we show that an inference system for partial higher-order conditional specifications, which is equationally complete w.r.t. the class of all extensional models, can be obtained from any equationally complete inference system for partial conditional specifications. Then, applying some previous results, we propose a deduction system, equationally complete for the class of extensional models of a partial conditional specification.*
Finally, turning the attention to the special important case of term-extensional models, we first show a sound and equationally complete inference system and then give necessary and sufficient conditions for the existence of free models, which are also free in the class of term-generated extensional models.

Introduction

Higher-order functions are now recognized as an important tool for the modular development of correct software systems. When the systems are developed by refinement from abstract specifications, it is rather natural to start with algebraic higher-order specifications. This higher-order approach, more or less explicit in the CIP method, is now at the basis of more recent projects, for example PROSPECTRA (see [K-B], also for further references).

Because of this solid practical motivation, a lot of work has been devoted recently to higher-order algebraic specifications (see, e.g. [M, MTW1, MTW2, Me, Q]). However little or nothing has been done for the algebraic specification of partial higher-order functions. Still in almost all real applications partial functions arise, especially in the abstract specification phase, when many details, including error messages, are not fixed.

This paper is an attempt at clarifying some basic hot points about partial higher-order specifications. We concentrate our attention on three classical issues: closure properties of the model classes, equational deduction and existence of free (and initial) models. It will turn out that in the partial case the situation is much more delicate then in the total one (see [Me] and [Q] for the same issues).

Following a classical approach (also adopted in [MTW1] and [Me], [Q]), we reduce higher-order specifications to special first-order specifications by introducing *apply* functions, one for each functional sort, and by restricting the class of models to *extensional* models, ie to the models where two functions giving the same result over any possible input are equal. So we are investigating classes of partial algebras satisfying not only the explicit axioms, but also the implicit *extensionality* axiom

$$* \qquad \forall\, f, g \quad ((\forall\, x{:}\ \mathit{apply}(f,x) = \mathit{apply}(g,x)) \supset f = g).$$

The extensionality axiom differs from the usual positive conditional axioms (the most frequently used in algebraic approaches, being the widest class guaranteeing the existence of initial (free) models, see e.g. [T,W,BW,B,R]) because of the internal quantification, which only involves the premises, and since the equality $f(x) = g(x)$ in the premises is *strong*, ie it holds iff either both sides are undefined or both denote the same element of the algebra, and not *existential*, ie holding iff both sides are defined and denote the same element.

In section 1 we show that, as in the total case, *the model class is not closed w.r.t. subobjects*; but, differently from the total case, *the closure w.r.t. products is lost*, too, so that *in general a partial conditional higher-order specification does not admit free models for any family of variables of arbitrary cardinality.*

This work has been partially supported by Esprit-BRA-WG COMPASS and by MURST 40%

In the total case an equationally complete inference system may be obtained by adding to an equationally-complete first-order system some extensionality rules, one for each arity; for unary functions it takes the form

$$\frac{f(x) = g(x)}{f = g} \qquad \text{f, g terms of functional sort , x variable} \notin \mathrm{Var}(f) \cup \mathrm{Var}(g)$$

The proof of the completeness of the enriched system (see e.g.[Me]) relies on the existence of free objects for a sufficiently high cardinality of the family of variables.

Instead in the partial case *an equationally complete inference system enriched by the extensionality rules is not equationally complete* (see fact. 2.4); but we get (section 2) an equationally complete system for the extensional models of any partial higher-order conditional specification starting from an equationally complete system for (first-order) partial conditional specifications by using a skolemization procedure.

In computer science a special role is played by term-generated (or reachable) models; thus (section 3) we analyze the subclass of extensional term-generated models of a partial conditional higher-order specification. Since every element of a term-generated algebra A is the evaluation of a term, the extensionality axiom holds in A iff A satisfies the following axiom, which, note, is an infinitary (countable) conditional axiom:

$$** \qquad \{f(t) = g(t) \mid t \in T_\Sigma\} \supset f = g,$$

The axiom **, called "term-extensionality axiom", characterizes the class of term-extensional algebras, which is smaller than the class of extensional algebras but includes all term-generated extensional models so that the subclass of all term-generated extensional models coincides with the subclass of all term-generated term-extensional models.

Since the class of all term-extensional models is the model class of a partial conditional specification, on the basis of some results in [AC1,AC2], we may obtain *an equationally complete system* for this class, together with *necessary and sufficient conditions for the existence of free models in the class of term-extensional models.*

1 Extensional models

We reduce, as in [MTW1], higher-order specifications to particular classes of first-order specifications and consider the class of extensional models of a conditional specification Sp, which in general does not admit initial nor free models. Since the class of extensional models is not closed under sub-objects, the usual correspondence between the existence of an initial model in the whole model class and in the subclass of the term-generated models is missing.

1.1 Partial higher-order specifications

We assume known the notions of signature, partial algebra and evaluation of terms (see e.g. [B, R, AC1]).

Partial conditional specifications. Let $\Sigma = (S,F)$ be a signature; the homomorphisms, as quite standard in the initial approach (see e.g. [B] and total Σ-homomorphisms in [BW]), are chosen in a way that the initial model, if any, satisfies the conditions of *no-junk* and *no-confusion*; let us recall the definition.
Let A and B be two Σ-algebras and p be a family of total functions $p = \{p_s\}_{s\in S}$, s.t. p_s: $s^A \to s^B$. Then p is a *homomorphism* from A into B iff for any $op \in F_{(s_1 \dots s_n, s_{n+1})}$, $n \geq 0$, and any $a_i \in s_i^A$, $i=1\dots n$,
$op^A(a_1,\dots,a_n) \in s_{n+1}^A$ implies $p_{s_{n+1}}(op^A(a_1,\dots,a_n)) = op^B(p_{s_1}(a_1),\dots,p_{s_n}(a_n))$.

Let X be a family of S-sorted variables.

- The set of all *elementary formulas* over Σ and X, denoted by EForm(Σ,X), consists of
  $\mathrm{EForm}(\Sigma,X) = \{D(t) \mid t \in T_\Sigma(X)_{|s}, s \in S\} \cup \{t = t' \mid t,t' \in T_\Sigma(X)_{|s}, s \in S\}$
  where D is called the *definedness predicate*.
- We denote by D(X) the set $\{D(x) \mid x \in X\}$.
- The set of all *conditional formulas* over Σ and X, denoted by CForm(Σ,X), consists of
  $\mathrm{CForm}(\Sigma,X) = \{\Delta \supset \varepsilon \mid \Delta \subseteq \mathrm{EForm}(\Sigma,X), \varepsilon \in \mathrm{EForm}(\Sigma,X)\}$.

If Δ is the empty set, then $\Delta \supset \varepsilon$ is an equivalent notation for the elementary formula ε.

- A *positive conditional formula* over Σ and X is a conditional formula $\Delta \supset \varepsilon \in CForm(\Sigma,X)$ s.t. for every $(t = t') \in \Delta$, $D(t) \in \Delta$ or $D(t') \in \Delta$.
- If $A \in PA(\Sigma)$, $\varphi \in CForm(\Sigma,X)$ and V is a (total) valuation for $Var(\varphi)$ (the S-sorted family of the variables occurring in φ) in A, then $A\ ‘_V\ \varphi$ is defined by:
  - $A\ ‘_V\ D(t)$ iff $t^{A,V} \in s^A$; $A\ ‘_V\ t = t'$ iff either $t^{A,V}, t'^{A,V} \notin s^A$ or $t^{A,V}, t'^{A,V} \in s^A$ and $t^{A,V} = t'^{A,V}$;
  - let φ be $\Delta \supset \varepsilon$; then $A\ ‘_V\ \varphi$ iff $A\ ‘_V\ \varepsilon$, or $A\ ’_V\ \delta$ for some $\delta \in \Delta$;

  We write $A\ ‘\ \varphi$ for a formula φ and say that φ *holds* in (equivalently*:* is *satisfied by, is valid in*) A iff $A\ ‘_V\ \varphi$ for all valuations V for $Var(\varphi)$ in A.
- If $C \subseteq PA(\Sigma)$ and $\varphi \in CForm(\Sigma,X)$, then $C\ ‘\ \varphi$, iff $A\ ‘\ \varphi$ for all $A \in C$. □

Remarks.

1 The equality that we consider is the so called *strong equality*, as opposed to the *existential equality* $t =_e t'$ ($A\ ‘_V\ t =_e t'$ iff $t^{A,V}$ and $t'^{A,V}$ are both defined and equal). Note that $D(t)$ stands for $t =_e t$.

2 The above notion of validity is the usual one in the many-sorted case; however some comments can be helpful. If $Var(\varphi)_s \neq \emptyset$ and $s^A = \emptyset$, then $A\ ‘\ \varphi$ holds; hence for any $C \subseteq PA(\Sigma)$, $C\ ‘\ \varphi$ iff $A\ ‘\ \varphi$ for all $A \in C$ s.t. $Var(\varphi)_s \neq \emptyset$ implies $s^A \neq \emptyset$. Thus if C contains an algebra with all supports non-empty (as it will always happens in the sequel), then the notion of validity for the class coincides with the classical one; for example we could not have both $C\ ‘\ \varphi$ and $C\ ‘\ \neg\varphi$ (but note that here we do not have negation). Finally it is also useful to emphasize that here we can stay within a two-valued logic, since a conditional formula for a total valuation of the variables is always either true or false.

3 Note that $A\ ‘\ \varepsilon$ implies $A\ ‘\ D(X) \supset \varepsilon$ for any X; if $X \subseteq Var(\varepsilon)$, then also the converse holds. Moreover, since $A\ ‘\ D(x)$ for any variable x and any A, the presence of $D(x)$ in $D(X) \supset \varepsilon$ has the only effect of possibly increasing the variables of the formula and thus the domain of the variable valuations.

Def. 1.1.

- A (*positive*) *conditional specification* consists of a signature Σ and of a set Ax of (positive) conditional formulas over Σ. A generic conditional specification will be denoted by Sp; the formulas belonging to Ax are called the *axioms* of Sp and denoted by (possibly decorated) α.
- For any conditional specification $Sp = (\Sigma,Ax)$, $Mod(Sp) = \{A \mid A \in PA(\Sigma), A‘\alpha\ \forall\ \alpha \in Ax\}$; an algebra $A \in Mod(Sp)$ is called a *model* of Sp. The class $GMod(Sp)$ consists of all term-generated models of Sp. □

For any conditional specification Sp the class $Mod(Sp)$ is not empty, since the trivial (total) algebra, with singleton sets as carriers and the obvious (total) interpretation of function symbols, is always a model.

Higher-order specifications. We define higher-order specifications as a special class of first-order specifications.

Def. 1.2.

- If S is a set of *basic sorts*, then the set $S^{\rightarrow}$ of *functional sorts* over S is inductively defined by: $S \subseteq S^{\rightarrow}$ and if $s_1,\dots,s_n,s_{n+1} \in S^{\rightarrow}$, then $s = (s_1 \times \dots \times s_n \rightarrow s_{n+1}) \in S^{\rightarrow}$ for all $n \geq 1$.

  A subset $S' \subseteq S^{\rightarrow}$ is *downward-closed* iff $s_1,\dots,s_n,s_{n+1} \in S'$ for all $(s_1 \times \dots \times s_n \rightarrow s_{n+1}) \in S'$.
- A *higher-order signature* $F\Sigma$, from now on h.o. signature, is a signature (S,F), where S is a downward-closed set of functional sorts, s.t. for all $s = (s_1 \times \dots \times s_n \rightarrow s_{n+1}) \in S$ with $n \geq 1$ there exists a distinguished operator $apply_s \in F_{(s\ s_1 \dots s_n, s_{n+1})}$. We will often use the infix notation for the $apply_s$ operators, ie we will write $f(a_1,\dots,a_n)$ for $apply_s(f,a_1,\dots,a_n)$, dropping the sort indexes when there is no ambiguity. Moreover we will not explicitly mention the apply functions in the definitions of concrete functional signatures.
- Let $F\Sigma = (S,F)$ be a h.o. signature; then $A \in PA(F\Sigma)$ is an *extensional partial algebra* iff satisfies the following *extensionality condition*:

  for all $s = (s_1 \times \dots \times s_n \rightarrow s_{n+1}) \in S$, with $n \geq 1$ and for all $f,g \in s^A$,

  if for all $a_i \in s_i^A$, $i=1,\dots,n$, $f(a_1,\dots,a_n) = g(a_1,\dots,a_n)$, then $f = g$.

An extensional partial algebra is called an E-algebra. We denote by $EPA(F\Sigma)$ the class of all E-algebras on $F\Sigma$.

- A *(positive) conditional higher-order specification* $(P)FSp = (F\Sigma,Ax)$ consists of a higher-order signature $F\Sigma$ and a set Ax of (positive) conditional axioms over $F\Sigma$.

  A generic (positive) higher-order specification will be denoted by $(P)FSp$. The class of *extensional models* of FSp, denoted by $EMod(FSp)$, is $Mod(FSp) \cap EPA(F\Sigma)$; while $EGMod(FSp)$ is the class of extensional term-generated models, i.e. $GMod(FSp) \cap EPA(F\Sigma)$. □

Note that for any h.o. signature $F\Sigma = (S,F)$, S is required to be downward closed in order that the operators $apply_s$ have arity in $S^* \times S$.

Remark. Any $A \in EPA(F\Sigma)$ is isomorphic to an algebra where the carriers of higher-order sort $(s_1 \times \ldots \times s_n \to s_{n+1})$ are subsets of the space of the partial functions from $s_1^A \times \ldots \times s_n^A$ into s_{n+1}^A and the $apply_s$ operators are interpreted in the standard way. Therefore in the following examples we assume that the higher-order carriers are function spaces and that the $apply_s$ functions are interpreted accordingly.

1.2 Counter-examples

Def. 1.3. Let A be a partial algebra on a signature $\Sigma = (S,F)$; then a Σ-algebra B is a *subalgebra* (regular subobject) of A iff $s^B \subseteq s^A$ $\forall\, s \in S$ and op^B is the restriction of op^A to $s_1{}^B \times \ldots \times s_n{}^B$ $\forall$ $op \in F_{(s_1 \ldots s_n,s)}$. □

Analogously to the case of higher-order *total* algebras, the class of all extensional algebras is not closed w.r.t. subobjects so that, in particular, the class of all extensional algebras cannot be expressed as the model class of a conditional specification, because the model class of any conditional specification is closed under subobjects (see e.g. [AC2] prop.1.3). But while in the total case the extensional algebras are closed w.r.t. non-empty direct products (of course performed in the class of *all* algebras), as claimed for example by the theorem 5.3 in [Me], in the partial frame also this closure is missing.

Fact 1.4. Let $F\Sigma$ be a h.o. signature; in general $EPA(F\Sigma)$ *is not closed w.r.t. subobjects*, *nor w.r.t. non-empty direct products.*

Proof. Consider the following example.

Signature $F\Sigma$ Sorts: s, $(s \to s)$ Operations: $f,g \colon \to (s \to s)$

Consider the algebras A , B and C, defined by:

$s^A = \{\bullet\}$; $(s \to s)^A = \{\bot, Id\}$, $\bot(\bullet)$ is undefined, $Id(\bullet) = \bullet$, $f^A = Id$; $g^A = \bot$;
$s^B = s^A$; $(s \to s)^B = (s \to s)^A$ $f^B = \bot$; $g^B = Id$;
$s^C = \emptyset$; $(s \to s)^C = (s \to s)^A$ $f^C = f^A$; $g^C = g^A$.

Then obviously $A,B \in EPA(F\Sigma)$, while $C \notin EPA(F\Sigma)$ and C is a subalgebra of A, by definition. Therefore $EPA(F\Sigma)$ is not closed w.r.t. subobjects.

Let us define $A \times B$: $s^{A\times B}=\{(\bullet,\bullet)\}$; $(s \to s)^{A\times B}=(s \to s)^A \times (s \to s)^B$; $f^{A\times B}=(f^A,f^B)$; $g^{A\times B}=(g^A,g^B)$.

Thus $(s \to s)^{A\times B}$ has cardinality four, while there are just two distinct partial functions, the identity and the totally undefined function, from $s^{A\times B}$ into $s^{A\times B}$, because $s^{A\times B}$ has cardinality one. Therefore $A\times B \notin EPA(F\Sigma)$ and hence $EPA(F\Sigma)$ is not closed w.r.t. non-empty direct products. □

Def. 1.5. Let X be a family of variables and $\mathcal{C}$ be a class of Σ-algebras. A couple (Fr,m), where $Fr \in \mathcal{C}$ and m is a valuation for X in Fr, is *free* over X in $\mathcal{C}$ iff

$\forall\, A \in \mathcal{C}$, $\forall\, V \colon X \to A$ there exists a unique homomorphism $p_V \colon Fr \to A$ s.t. $p_V(m(x)) = V(x)$ $\forall$ $x \in X$.

An algebra I is initial in $\mathcal{C}$ iff it is free over the empty family of variables in $\mathcal{C}$, ie iff $I \in \mathcal{C}$ and for all $A \in \mathcal{C}$ there exists a unique homomorphism from I into A. □

It is easy to see that initial and terminal algebras in $PA(F\Sigma)$ are also extensional and hence initial and terminal in $EPA(F\Sigma)$; but, although $EPA(F\Sigma)$ has an initial model, in general both the class of all extensional models and the class of all term-generated models for equational specifications have no initial model.

Fact 1.6. Let $F\Sigma = (S,F)$ be a higher-order signature and FSp be an equational specification $(F\Sigma,Ax)$. Then *in general there does not exist an E-algebra initial in* EMod(FSp) *nor in* EGMod(FSp).

Proof. Consider the following example.

Specification FSp_1

Sorts: $s, (s \to s)$ Operations: $e: \to s$ Axioms: α_1 D(f)

$f,g: \to (s \to s)$ α_2 D(g)

Then proceed by contradiction assuming that there exists I initial in EMod(FSp_1) (resp. in EGMod(FSp_1)).

Let F and G be the E-algebras defined by:

$s^F = \{\bullet\}$; $(s \to s)^F = \{\bot, Id\}$, $\bot(\bullet)$ is undefined, $Id(\bullet) = \bullet$ $e^F = \bullet$ $f^F = Id$; $g^F = \bot$

$s^G = s^F$; $(s \to s)^G = (s \to s)^F$ $e^G = \bullet$ $f^G = \bot$; $g^G = Id$.

Both F and G belong obviously to EGMod(FSp_1); thus, because of the initiality of I, there exist two homomorphisms $p^F: I \to F$ and $p^G: I \to G$. Then it is just routine to show that the existence of such p^F and p^G implies that for all $a \in s^I$ both $f^I(a)$ and $g^I(a)$ are undefined and hence that $f^I = g^I$, because of extensionality; thus we get $g^F = p^F(g^I) = p^F(f^I) = f^F$, in contradiction with the definition of f^F and g^F. □

The above example suggests that for the existence of the initial model, the minimal *definedness* may conflict with the minimal *equality*. Indeed if the elements in the domain are too few, then we cannot distinguish the functions and hence the minimal definedness (on the arguments) may force the *maximal* equality (on the functions). For the same reason we have that two functions having the same result over every tuple of terms because of the axioms, may differ on some *non-term-generated* argument-tuple, so that the equalities between ground terms holding in the term-generated models may be strictly more than the equalities holding in all models. In particular the equalities between ground terms holding in all the term-generated models may define an extensional algebra, so that there exists an initial model in EGMod(FSp), while the equalities between ground terms holding in all models are too few.

Fact 1.7. Let $F\Sigma = (S,F)$ be a h.o. signature and FSp be an equational specification $(F\Sigma,Ax)$ s.t. I is initial in EGMod(FSp). Then *in general* I *is not initial in* EMod(FSp) and the sets $\{\varepsilon \mid \varepsilon \in EForm(F\Sigma,\emptyset)$, EMod(FSp)‘$\varepsilon\}$ and $\{\varepsilon \mid \varepsilon \in EForm(F\Sigma,\emptyset)$, EGMod(FSp))‘$\varepsilon\}$ are different.

Proof. Consider the following example.

Specification FSp_2

Sorts: $s_1, s_2, (s_1 \to s_2)$ Operations: $e: \to s_1$ Axioms: α_1 D(f(e))

$f,g: \to (s_1 \to s_2)$ α_2 $f(e) = g(e)$

Then all term-generated models are isomorphic to I, defined by:

$s_1{}^I = \{\bullet\}$; $s_2{}^I = \{\bullet\}$; $(s_1 \to s_2)^I = \{Id\}$, where $Id(\bullet) = \bullet$; $e^I = \bullet$; $f^I = g^I = Id$.

So that I is initial in EGMod(FSp_2); however I is not initial in EMod(FSp_2), since there are (no term-generated) models A for which $f^A \neq g^A$. Moreover EGMod(FSp_2) ‘ f = g, while EMod(FSp_2) ’ f = g, because A ’ f = g. □

In the total case if a family X of variables has a sufficiently high cardinality, then there exists the free model on X in the class of all extensional models of a conditional specification (see theorems 3.7 and 5.7 of [Me]). Instead in the partial case there are conditional specifications whose classes of extensional models do not admit free models for families of variables of arbitrary cardinality.

Fact 1.8. Let $F\Sigma = (S,F)$ be a h.o. signature, FSp be an equational specification $(F\Sigma,Ax)$ and X be a family of variables of arbitrary cardinality. Then *in general there does not exist a free model for* X *in* EMod(FSp).

Proof. Consider again the specification FSp_1 and the algebras F and G defined in fact 1.6; we show that there does not exist a free model for X in EMod(FSp_1).

Assume by contradiction that (I,m) is free in EMod(FSp_1) for a family X of variables. Let $V^F: X \to F$ and $V^G: X \to G$ be any valuations, which always exist, because F and G have all the carriers non-empty.

Because of the freeness of I, there exist two homomorphisms $p^F: I \to F$ and $p^G: I \to G$ s.t. $p^F \cdot m = V^F$ and $p^G \cdot m = V^G$. Thus, as in in fact 1.6, we get $g^F =_e p^F(g^I) =_e p^F(f^I) =_e f^F$, in contradiction with the definition of f^F and g^F. □

Note that the above counter-example also applies to the subclass of extensional models generated by the family X of variables, EGMod(FSp,X) = $\{A \mid A \in \text{EMod(FSp)}, \exists\, V:X \to A \text{ s.t. } eval^{A,V}(T_\Sigma(X)) = A\}$, because F and G, being term-generated, belong to EGMod(FSp,X).

2 Equational deduction

The focus of logic deduction in the total algebraic case is on *equational* deduction, because an inference system complete w.r.t. the equations gives the (initial) free model. In the partial case only the definedness and the equality between defined terms (ie existential equalities) are needed in order to characterize the (initial) free model, if any (see eg [BW]). Here we also need to consider conditional axioms with strong equalities in the premises and hence we deal also with strong equalities. Moreover our deduction is sound and complete not only w.r.t. equalities, but also w.r.t. formulas of the form $D(X) \supset \varepsilon$ which corresponds to the formula $\forall\, (X \cup Var(\varepsilon)).\ \varepsilon$, with explicit quantification, considered in the many sorted total case (see [MG]) both for clarifying equational deduction and for obtaining models free w.r.t. a family of variables. Hence we give notions of soundness and completeness also dealing with such particular conditional formulas; our notions subsume the usual ones only dealing with equalities.

Def. 2.1. Let Sp be a conditional specification, *C* a subclass of Mod(Sp) and L an inference system.
L is *sound* for *C* iff L “ φ implies *C* ‘ φ for all conditional formulas φ.
L is *strongly complete* (for complete w.r.t. strong equalities) for *C* and a family of variables X iff for all $\varepsilon \in EForm(\Sigma,X)$ and all $Y \subseteq X$ *C* ‘ $D(Y) \supset \varepsilon$ implies L “ $D(Y) \supset \varepsilon$. □

Note that if L is strongly complete for *C* and X, then in particular *C* ‘ ε implies L “ ε.

In the total case (see [Me,Q]) a complete system for the class of extensional models may be obtained by enriching a complete system for the whole class of algebras by extensionality rules, one for each arity; for example for unary functions the rule takes the form

* $\dfrac{f(x) = g(x)}{f = g}$ f, g terms of functional sort $s \to s'$, $x \in X_s$, $x \notin Var(f) \cup Var(g)$

Instead in the partial case the above rule * is insufficient to achieve a complete system. To propose an example of this claim and also for further use let us recall the definition of the system CL(Sp), from [AC2], which is sound and strongly complete (see theorem 3.11 in [AC2]).

Def. 2.2. The CL(Sp) system for a conditional specification $Sp = (\Sigma,Ax)$ consists of the axioms Ax and of the following axiom schemas and inference rules, where we assume that as usual $\varepsilon \in EForm(\Sigma,X)$, $\Delta,\Delta_\gamma,\Gamma,\Theta_j,\Gamma_j$ are countable subsets of EForm(Σ,X), $x \in X$ and $t,t',t'',t_i,t'_i,t_x \in T_\Sigma(X)$.

0 $D(x)$ *Definedness of variables*

1 $t = t$ *Congruence*
2 $t = t' \supset t' = t$
3 $\{t = t', t' = t''\} \supset t = t''$
4 $\{t_i = t'_i \mid i=1\ldots n\} \supset op(t_1,\ldots,t_n) = op(t'_1,\ldots,t'_n)$

5 $D(op(t_1,\ldots,t_n)) \supset D(t_i)$ *Strictness*

6 $\{D(t), t = t'\} \supset D(t')$ *Definedness and equality*

7 $\dfrac{\Delta \cup \Gamma \supset \varepsilon,\ \{\Delta_\gamma \supset \gamma \mid \gamma \in \Gamma\}}{D(Var(\Gamma) - Var(\cup_{\gamma\in\Gamma} \Delta_\gamma \supset \varepsilon)) \cup \Delta \cup (\cup_{\gamma\in\Gamma} \Delta_\gamma) \supset \varepsilon}$ *Modus Ponens*

8 $\dfrac{\Delta \supset \varepsilon}{\{D(t_x) \mid x \in X\} \cup \{\delta[t_x/x \mid x \in X] \mid \delta \in \Delta\} \supset \varepsilon[t_x/x \mid x \in X]}$ *Instantiation/Abstraction*

$$9 \quad \frac{\{\Theta_j \cup \Gamma_j \supset \varepsilon \mid j \in J\}}{D(\bigcup_{j \in J} Var(\Gamma_j)) \cup (\bigcup_{j \in J} \Theta_j) \supset \varepsilon}$$

Elimination
$\forall \{\gamma_j\}_{j \in J}$ with $\gamma_j \in \Gamma_j$ $\exists$ t,t' s.t. $D(t), D(t'), t=t' \in \{\gamma_j\}_{j \in J}$.

If the axioms of Sp are finitary (only a finite number of elementary formulas in the premises), then rules 7, 8 and 9 can be replaced by the rules

$$7_f \quad \frac{\Delta \cup \{\gamma\} \supset \varepsilon,\ \Delta_\gamma \supset \gamma}{D(Var(\gamma) - Var(\Delta_\gamma)) \cup (\Delta \cup \Delta_\gamma) \supset \varepsilon} \qquad 8_f \quad \frac{\Delta \supset \varepsilon}{\{D(t)\} \cup \{\delta[t/x] \mid \delta \in \Delta\} \supset \varepsilon[t/x]}$$

$$9_f \quad \frac{\Delta_1 \cup \{D(t)\} \supset \varepsilon,\ \Delta_2 \cup \{D(t')\} \supset \varepsilon,\ \Delta_3 \cup \{t = t'\} \supset \varepsilon}{D(Var(t = t')) \cup (\Delta_1 \cup \Delta_2 \cup \Delta_3) \supset \varepsilon}$$

where all sets of elementary formulas are finitary and in this case the system is called $CL_f(Sp)$. □

Remarks. Two comments are in order here.

1 Notice how the well-known empty-carrier problem is handled here (see [MG] and recall that an [MG]-like formula $\forall(X \cup Var(\Delta \supset \varepsilon)).\Delta \supset \varepsilon$ is represented by $D(X) \cup \Delta \supset \varepsilon$). We can eliminate D(x) from the premises of a formula $\forall(X \cup Var(\Delta \supset \varepsilon)).\Delta \supset \varepsilon$ only if either D(x) is redundant, because x already appears in $\Delta \supset \varepsilon$, or x does not appear in $\Delta \supset \varepsilon$ and there exists a *defined* ground term of the right sort to instantiate x. Indeed a premise may be eliminated only by rule 7 (modus ponens) by which the variables of a formula do not decrease; thus if x already appears in $\Delta \supset \varepsilon$, then applying rules 7 and 0 to $D(X) \cup \Delta \supset \varepsilon$ we can deduce $D(X-\{x\}) \cup \Delta \supset \varepsilon$; otherwise if x does not appear in $\Delta \supset \varepsilon$ and there exists a defined ground term t to instantiate x, then from rule 8 we get $D(X-\{x\}) \cup \{D(t)\} \cup \Delta \supset \varepsilon$ and hence from rule 7 and D(t) we conclude $D(X-\{x\}) \cup \Delta \supset \varepsilon$. But if x does not appear in $\Delta \supset \varepsilon$ and there does not exist a defined ground term to instantiate x, then there is no way to eliminate D(x) from the premises.

2 The elimination rule 9 is better understood as a generalization of the corresponding rule 9_f for the finitary case, which is rather simple and intuitive (though the proof that 9_f can replace 9 is quite difficult; see theorem 4.2 of [AC2]). Forgetting the definedness of variables, it is an inference rule which can be deduced in first-order logic with negation and disjunction (which we here not have): e.g. for $\Delta_1 = \Delta_2 = \Delta_3 = \emptyset$, the premises are $\neg D(t) \vee \varepsilon$, $\neg D(t') \vee \varepsilon$, $\neg t=t' \vee \varepsilon$ from which we get $(\neg D(t) \wedge \neg D(t') \wedge \neg t=t') \vee \varepsilon$, and since $\neg D(t) \wedge \neg D(t') \supset t=t'$, we get ε. Moreover it is straightforward to see 7_f and 8_f as particular cases of 7 and 8 in the finitary case. Note that in this section if we restrict ourselves to h.o. specifications with finitary axioms, then we can use the system for the finitary case; later on in section 3 it will be instead essential to use the system for the infinitary case, because of an implicit infinitary rule corresponding to the extensionality axiom for term-extensional models.

Theorem 2.3. Let Sp = (Σ,Ax) be a conditional specification [s.t. all the axioms in Ax are finitary] and X be a [finitary] family of variables. Then CL(Sp) [$CL_f(Sp)$] is sound and strongly complete for Mod(Sp) and X. □

Fact 2.4. Let FSp be a conditional higher-order specification. Then the system, from now on denoted by FSp “, consisting of all the axiom schemas and inference rules of CL(FSp) and of the following further inference rules:

$$10 \quad \frac{f(x_1,\ldots,x_n) = g(x_1,\ldots,x_n)}{f = g}$$

$x_i \in X_{s_i}$, $x_i \notin Var(f) \cup Var(g)$, i=1...n
$f,g \in T_\Sigma(X)|_{s_1 \times \ldots \times s_n \to s}$

is not complete for EMod(FSp) and the empty family of variables.

Proof. Consider the following specification FSp = (FΣ,Ax), defined by:

Sorts	s, (s → s)	Axioms	α_1: f = g ⊃ D(e)	
Operations	f,g: → (s → s)		α_2: D(f(x)) ⊃ D(e)	α_3: D(g(x)) ⊃ D(e)
	e: → s		α_4: D(f)	α_5: D(g)

Then e is defined in each model A of FSp; indeed either there exists an element a s.t. $f^A(a)$ or $g^A(a)$ is defined, and in this case because of α_2 and α_3 also e^A is defined, or both f^A and g^A are defined (because of α_4 and α_5) and their result over any possible assignement is undefined so that, because of the extensionality, $f^A = g^A$ and hence D(e) follows from α_1. But it easy to check that FSp ” D(e). □

Although rule 10 is insufficient to make the system CL complete, we can obtain a strongly complete inference system for the class of extensional models from any strongly complete inference system by applying a technique of skolemization; let us introduce the basic scheme of this translation before stating formally the result. To do this we informally use the full first-order language based on EForm(FΣ,X), where the validity is defined in the obvious way.

Let $F\Sigma = (S,F)$ be a h.o. signature and FSp be a higher-order conditional specification $(F\Sigma,Ax)$.
Then EMod(FSp) is the class of all (usual) models of FSp satisfying the non-conditional axioms $\alpha_s = \{\forall f,g: s.[\forall x_1:s_1,\dots,\forall x_n: s_n.\ f(x_1,\dots,x_n) = g(x_1,\dots,x_n)] \supset f = g\}$ for all $s = s_1\times\dots\times s_n \rightarrow s_{n+1}$ in S. In order to have a complete deduction system w.r.t. EMod(FSp), we first reduce the α_s to conditional axioms, by a usual logical procedure of skolemization. Let us consider for simplicity unary functions, ie let s be $s' \rightarrow s''$; then α_s is logically equivalent to $\forall f,g:s.(\neg[\forall x:s'.f(x)=g(x)] \vee f=g)$ and then to $\forall f,g:s.([\exists x:s'.\neg f(x)=g(x)] \vee f=g)$. By using Skolem functions we reduce the last formula to $\forall f,g: s.[\neg f(x(f,g)) = g(x(f,g))] \vee f = g$ and finally this one is equivalent to $\beta_s = \{\forall f,g:s.f(x(f,g))=g(x(f,g)) \supset f=g\}$. Since skolemization preserves satisfiability, for any conditional formula φ on FΣ there exists $A\in$ EMod(FSp) which does not satisfy φ, ie A ‘ $Ax \wedge \{\alpha_s \mid s\in S\} \wedge \neg\varphi$, iff there exists $B\in$ Mod(FSp') which does not satisfy φ, where $FSp' = (F\Sigma,Ax) \cup (\cup_{s=(s_1\times\dots\times s_n\rightarrow s_{n+1})\in S} (\{x_i: s\times s \rightarrow s_i \mid i=1,\dots,n\},\{\beta_s\}))$.
Therefore any strongly complete deduction system for FSp' is a strongly complete deduction system for FSp, too.

Note that the axioms not in Ax are finitary.

Def. 2.5. Let $F\Sigma = (S,F)$ be a h.o. signature and FSp be the conditional specification $(F\Sigma,Ax)$. We denote by SK(FSp) the conditional specification (SK(FΣ),SK(Ax)), where

- $SK(F\Sigma) = (S,F\cup(\{x_i:s\times s\rightarrow s_i \mid i=1,\dots,n\}_{s=(s_1\times\dots\times s_n\rightarrow s_{n+1})\in S})$
- $SK(Ax) = Ax \cup \{f(x_1(f,g),\dots,x_n(f,g)) = g(x_1(f,g),\dots,x_n(f,g)) \supset f = g\}_{s=(s_1\times\dots\times s_n\rightarrow s_{n+1})\in S}$. □

Theorem 2.6. Let FSp be a partial higher-order conditional specification and X be a family of variables. Every sound strongly complete system for Mod(SK(FSp)) and X, is a sound and strongly complete system for EMod(FSp) and X. □

Corollary 2.7. Let FSp = (FΣ,Ax) be a higher-order conditional specification [s.t. all the axioms in Ax are finitary] and X be a [finitary] family of variables. Then CL(SK(FSp)) [CL_f(SK(FSp))] is sound and strongly complete for EMod(FSp) and X. □

Theorem 2.8. Let FSp = (FΣ,Ax) be a higher-order conditional specification X be a family of variables and L be a sound and strongly complete system for EMod(FSp) and X. An FΣ-algebra F is free over X in EMod(FSp) iff it is isomorphic to $T_{F\Sigma}(X)/{\equiv_L}$, where $\equiv_L$ is the congruence

$\{(t,t') \mid t,t'\in T_{F\Sigma}(X),\exists\ Y\subseteq X$ s.t. *L* “ $D(Y)\supset t=t'$,(*L* “ $D(Y)\supset D(t)$ or *L* “ $D(Y)\supset D(t'))\}$. fi

3 Term-extensional models

Although mathematical aspects may be more elegant if non-term-generated models are allowed and stepwise refinement is made easier, because extra-elements and structures may be added in a second moment, we cannot ignore that the computer science focus is on term-generated models; for example in [W] only (first-order) term-generated models, there called *computation structures*, are considered when defining abstract data types.

In the case of higher-order specifications, together with term-generated models we have also the interesting class of what we have called in [AC1] *term-extensional models*, i.e. the h.o. models where two functions are equal iff they give the same results when applied to tuples of term-generated arguments.

Partial specifications of term-extensional models can be seen as a special subclass of partial non-positive conditional specifications, which are studied in [AC1, AC2, C]. On the basis of these results we can obtain new results about equational deduction and existence of free models. For the class of term-extensional models it is possible to give directly, without skolemization, a strongly complete inference system. Moreover we can completely clarify the issue of free models and give necessary and sufficient conditions for their existence.

Def. 3.1. Let $F\Sigma = (S,F)$ be a h.o. signature.

- An $F\Sigma$-algebra A is *term–extensional* iff for any $f,g \in (s_1 \times \ldots \times s_n \rightarrow s_{n+1})^A$, $f(t_1{}^A,\ldots,t_n{}^A) = g(t_1{}^A,\ldots,t_n{}^A)$ for all $t_i \in T_{F\Sigma|s_i}$ and i=1...n implies f = g.
- Let FSp be the conditional higher-order specification $(F\Sigma,Ax)$. Then the class TEMod(FSp) is the class of all *term-extensional* models of FSp, ie $TEMod(FSp) = Mod(FSp^{ext})$, where $FSp^{ext} = (F\Sigma, Ax \cup Ax^{ext})$ and Ax^{ext} is the set $\{\{f(t_1,\ldots,t_n)=g(t_1,\ldots,t_n) \mid t_i \in T_{\Sigma|s_i}, i=1,\ldots,n\} \supset f=g \mid (s_1 \times \ldots \times s_n \rightarrow s_{n+1}) \in S\}$ and f, g are variables of sort $(s_1 \times \ldots \times s_n \rightarrow s_{n+1})$. □

Note that a term-generated algebra is extensional if and only if it is term-extensional, because every element is the evaluation of a ground term; so for any conditional higher-order specification FSp we have that in particular term-generated and term-extensional models are just the term-generated extensional models.

Equational deduction. Let us consider equational deduction in this special case of term-extensional models. First note that in order to get completeness it is not enough adding the rule

$$\frac{\{f(t) = g(t) \mid t \in T_{F\Sigma}\}}{f = g}$$

(see fact 2.4 for a motivation) which is a special case of the *infinitary induction* ω*-rule* we can use for making complete in GMod(Sp) any complete system for Mod(Sp) (see [W]). However since in this case higher-order specifications are reduced to particular non-positive conditional specifications, also inference systems for the higher-order case are particular inference systems for non-positive conditional specifications. Of course these systems are infinitary since Ax^{ext} contains infinitary conjunctions in the premises.

Def. 3.2. Let FSp be the conditional higher-order specification $(F\Sigma,Ax)$ and FSp^{ext} be the conditional specification $(F\Sigma, Ax \cup Ax^{ext})$. The system FCL(FSp) is the system $CL(FSp^{ext})$. □

Theorem 3.3. Let FSp be the conditional higher-order specification $(F\Sigma,Ax)$ and X be a family of variables. Then the conditional system FCL(FSp) is sound and strongly complete for TEMod(FSp) and X. □

Free models. The counter-example of fact. 1.6 shows that in general conditional higher-order specifications do not admit free and initial models in the class of all term-generated models.

The following result completely characterizes the existence of free models, giving necessary and sufficient conditions both in terms of semantic conditions and in terms of equational deduction.

Some informal comments may facilitate the understanding of the theorem. Condition 3 is a semantic condition, stating that if two functional terms are always defined, then either they are equal in all models or there exists a distinguishing tuple of arguments for them. Condition 4 is just the equivalent of condition 3 in terms of logical deduction and is an immediate consequence of the completeness of the system FL. Finally condition 5 is the specialization of the condition 4 to the complete system FLC(PFSp) that we have exhibited before.

It is interesting to note that starting from these conditions we can show that the existence of free models is undecidable (see [AC2]).

In the sequel by Gen(*C*, X), where *C* is a class of Σ-algebras and X a family of variables, we denote the subclass of *C* defined by $\{A \mid A \in C \text{ s.t. } \exists\, V: X \rightarrow A \text{ s.t. } eval^{A,V}(T_\Sigma(X)) = A\}$.

Theorem 3.4. Let $PFSp = (F\Sigma,Ax)$ be a positive conditional higher-order specification, X be a family of variables and FL be a strongly complete system for TEMod(PFSp) and X.

Then the following conditions are equivalent.

1. there exists a free object for X in TEMod(PFSp);
2. there exists a free object for X in Gen(TEMod(PFSp),X);
3. $\forall\, f,g \in T_{F\Sigma}(X)_{|(s_1 \times \ldots \times s_n \rightarrow s_{n+1})}$, $n \geq 1$, s.t. Gen(TEMod(PFSp),X)‘D(f) and Gen(TEMod(PFSp),X)‘D(g)

- either Gen(TEMod(PFSp),X) ‘ f = g,
- or there exist $t_i \in T_{F\Sigma|s_i}$, i=1,…,n, s.t. Gen(TEMod(PFSp),X) ’ $f(t_1,\dots,t_n) = g(t_1,\dots,t_n)$ and (Gen(TEMod(PFSp),X) ‘ $D(f(t_1,\dots,t_n))$ or Gen(TEMod(PFSp),X) ‘ $D(g(t_1,\dots,t_n))$);

4. ∀ f,g∈ $T_{F\Sigma}(X)_{|(s_1\times\dots\times s_n\rightarrow s_{n+1})}$, n ≥ 1, s.t. FL “ D(X) ⊃ D(f) and FL “ D(X) ⊃ D(g)
   - either FL “ D(X) ⊃ f = g,
   - or there exist $t_i \in T_{F\Sigma|s_i}$, i=1,…,n, s.t. FL ” D(X) ⊃ $f(t_1,\dots,t_n) = g(t_1,\dots,t_n)$ and (FL “ D(X) ⊃ $D(f(t_1,\dots,t_n))$ or FL “ D(X) ⊃ $D(g(t_1,\dots,t_n))$);
5. ∀ f,g∈ $T_{F\Sigma}(X)_{|(s_1\times\dots\times s_n\rightarrow s_{n+1})}$, n≥1, s.t. FCL(PFSp) “ D(X) ⊃ D(f) and FCL(PFSp) “ D(X) ⊃ D(g)
   - either FCL(PFSp) “ D(X) ⊃ f = g,
   - or ∃ $t_i \in T_{F\Sigma|s_i}$, i=1,…,n, s.t. FCL(PFSp) ” D(X) ⊃ $f(t_1,\dots,t_n) = g(t_1,\dots,t_n)$ and (FCL(PFSp) “ D(X) ⊃ $D(f(t_1,\dots,t_n))$ or FCL(PFSp) “ D(X) ⊃ $D(g(t_1,\dots,t_n))$). □

In the particular case of *total* functions the above condition 3 in the theorem 3.4 is always satisfied, so that in the class of all term-extensional total models there exists a free model for each family of variables.

Corollary 3.5. Let X be a family of variables and PFSp = (FΣ,Ax) be a positive conditional higher-order specification. Then there exists a free model for X in the class of total term-extensional models of PFSp. □

Conclusion. This paper presents some basic results about partial h.o. specifications, which illustrate in particular the difference with the total h.o. case. Some of the results in the paper can be further enlightened when seen as particular applications of the technique of simulation of institutions (see [AC3] and also [Mes] for a similar notion). In particular this technique can be used to partly illustrate the relationship between total and partial specification of h.o. partial functions. We have the feeling that adopting a total approach to partial functions with explicit values for undefinedness does not change the nature of difficulties. We are currently working on this relationship and hope to come out with a complete picture, but we have found that things are less easy than thought and misbeliefs abound in the folklore.

References

AC1 Astesiano, E.; Cerioli, M. “On the Existence of Initial Models for Partial (Higher-Order) Conditional Specifications”, Proc. TAPSOFT’89, vol.1, Lecture Notes in Computer Science n. 351, 1989.

AC2 Astesiano, E.; Cerioli, M. “Free Objects and Equational Deduction for Partial Conditional Specifications”, Tecnhical Report n.3, Formal Methods Group, University of Genova, 1990.

AC3 Astesiano, E.; Cerioli, M. “Commuting between Institutions via Simulation”, submitted, 1990.

B Burmeister, P. A Model Theoretic Oriented Approach to Partial Algebras, Berlin, Akademie-Verlag, 1986.

BW Broy, M.; Wirsing, M. “Partial abstract types”, Acta Informatica 18, 1982.

C Cerioli, M. “A sound and equationally-complete deduction system for partial conditional (higher order) types”, in Proc.3rd Italian Conference of Theoretical Computer Science,1989, Singapore, World Scientific.

GB Goguen J.A.; Burstall R.M. “Institutions: Abstract Model Theory for Specification and Programming”. Technical Report of Computer Science Laboratory, SRI International, 1990.

K-B Krieg-Brückner B. “Algebraic Specification and Functionals for Transformational Program and Meta Program Development”, in Proc.TAPSOFT'89, Lecture Notes in Computer Science n. 352, 1989.

M Möller, B. “Algebraic Specification with Higher-Order Operations”, Proc. IFIP TC 2 Working Conference on Program Specification and Transformation, North-Holland, 1987.

Me Meinke, K. “Universal Algebra in Higher Types” to appear in Theoretical Computer Science, 1990.

Mes Meseguer J. “General logic” in *Proc. Logic Colloquium ’87*, North-Holland, ’89.

MG Meseguer, J.; Goguen, J.A. “Initiality, Induction and Computability”, in Algebraic Methods in Semantics, Cambridge, Cambridge University Press, 1985.

MTW1 Möller B., Tarlecki A., Wirsing M. “Algebraic Specification with Built-in Domain Constructions”, in Proc. of CAAP ’88, Lecture Notes in Computer Science n.299, 1988.

MTW2 Möller B., Tarlecki A., Wirsing M. “Algebraic Specifications of Reachable Higher-Order Algebras”, in Recent Trends in Data Type Specification, Lecture Notes in Computer Science n.332, 1988.

Q Qian Z. “Higher-Order Order-Sorted Algebras”, Proc. 2nd International Conference on Algebraic and Logic Programming, Nancy Oct. 1990, Lecture Notes in Computer Science, Berlin, Springer-Verlag, 1990

R Reichel H. *Initial Computability, Algebraic Specifications, and Partial Algebras*, Berlin, Akademie-Verlag, 1986.

T Tarlecki A. “Quasi-varieties in Abstract Algebraic Institutions”, *Journal of Computer and System Science*, n. 33, 1986.

W Wirsing, M. “Algebraic Specification”, in Handbook of Theoretical Computer Science vol.B, North Holland, 1990.