COMPUTABILITY IN CATEGORIES

M.B. Smyth

Dept. of Computer Studies
University of Leeds
Leeds, England

-Extended Abstract-

0. INTRODUCTION

Questions of effectiveness have been considered in connexion with several of the special categories of domains studied in denotational semantics. Let us mention Tang [28] vis-a-vis Scott's "classical" category CL of continuous lattices [21], [22]; Egli and Constable [5] for the category ACPO of algebraic bounded-complete cpo's; Smyth [23] for the cateogry CCPO of continuous bounded-complete cpo's; Kanda [11] for Plotkin's SFP [19]. Each of these categories could lay some claim to being *the* category of interest in semantics. Recently, however, it has become clear that we cannot limit ourselves in advance to a small fixed set of categories of domains. Current work on parallelism, sequentiality and concrete domains, especially, gives rise to a proliferation of categories ([1], [4], [8]). Even apart from this proliferation, explicitly categorical constructions are playing an increasing role, especially in connexion with data types [6], [15]. The question arises, whether we can treat effectiveness itself in a categorical setting, so that the effectiveness of domains and functions can be defined in a uniform way, and - beyond that - the effectiveness of categorical notions themselves, especially that of a functor. In this paper we propose an affirmative answer, using (mainly) the notion of an *effectively given 0-category*.

Why should we give so much attention to effectiveness? One reason has to do with the systematic study of the power of specification techniques. We cannot require of a general purpose programming language that it be able to specify (define) all number-theoretic functions, but only (at most) those which are partial recursive. A corresponding distinction must be made for all the "data types" which one may wish to handle. And the problem is not simply that of picking out the computable functions over a given data type; we have the problem of specifying the data types themselves, and thus of determining the "computable", or effectively given, data types (i.e. the types which should in principle be specifiable). For data types considered simply as sets with operations, the discussion has reached a fairly advanced state (see [29] and references there given). For data types with *domains* as carriers (with which we are concerned here), the problems are much more complex. Little is known about the completeness or adequacy of specification methods for these types.

The requirements that all semantic constructs be effective can be a useful discipline, both negatively (thus, the power-domain construction of [14] remains suspect, since we cannot see how to make it effective), and as a positive aid in finding suitable constructs (see the remarks on "finite elements" in Sec. 1).

As our final reason for insisting on effectiveness, let us just cite the connexion between computing and mathematical constructivism, argued by Bishop [2] and Constable [3], and recently asserted in a very strong form by Martin-Löf [17].

Some of the proofs missing from this extended abstract can be found in [27].

1. OVERVIEW

The guiding principle of our work is a very simple one: in every system (domain, category, etc.) with which we have to deal, there is a clearly identifiable subsystem of finite elements, whose properties completely determine the whole system. The infinite elements have as it were a secondary existence as limiting processes applied to the finite elements (more concretely, as rules for the production of finite elements), all their properties being derivative from those of the finite elements. One naturally defines a *computable* (infinite) element as one which is given as an effective limit of finite elements (relative to a suitable enumeration of the finite elements).

This concentration on the finite elements was the main factor in the simplification of the power-domain constructs [24], and has also been fruitful in developing the theory of concrete domains [10]. Its main application in the present work is the characterization of the computable arrows/objects of a category as limit of chains of "finite" arrows/objects. Unfortunately we lapse from our principle in regard to effective(ly given) categories and functors themselves: we do not here build them from chains of finite categories and functors. On this point, see the concluding remarks of the paper.

In Section 2 we introduce the class of "admissible" categories, which forms the frameowrk of our investigation. An admissible category is essentially an O-category in which the subcategory of embeddings is algebraic (in the terminology of [24]). Some important properties of admissible categories are proved. In Section 3 we define the notion of *effective basis* of an admissible category, and characterize the *effectively given* objects and arrows in a category with effective basis. We indicate that these abstractly defined notions reduce to the usual ones when we specialize to the particular concrete categories in which effectiveness has been studied previously (we verify it in detail only for the case of ACPO). In Section 4 we introduce "frames" of functors as a counterpart to graphs of functions, and in terms of them define *computable* functors. We show that, for suitable indexings of our categories, this notion of computability of functors agrees with Kanda's in [11]; this means that we can use Kanda's results to get "fully effective" solutions of domain equations. The concluding section treates some miscellaneous topics: O-categories versus ω-categories; effective versions of the adjoint functor theorem; continuous versus algebraic domains; and a possible refinement of the notion of an effectively given category that seems to be required if we are able to construct functor categories effectively.

We have just alluded to Kanda's work on effective categories. The relationship between our work and that of Kanda has been discussed in [26]. Briefly, Kanda is not

concerned with developing a unified theory of computability; rather, he assumes that we have a countable category (thought of as a category of effective domains and computable functions), suitably indexed, already given to us. In terms of such indexed categories, the effectiveness of various categorial constructions (functors, initial fixpoints) can be handled straightforwardly. Our theory can be considered as an account of how the "suitably indexed" categories come about. The direct verification that given categories and functors are effective in Kanda's sense tends to be very complex; but, for categories/functors build up in the way we describe here, such verification is usually trivial.

2. ADMISSIBLE CATEGORIES

We begin with some definitions as in [30], [24], [26]:

DEFINITION 1. An O-*category* is a category in which (i) every hom-set is a partial order in which every ascending ω-chain has a lub, and (ii) composition of morphisms is ω-continuous with respect to these partial orders. If $A \xrightarrow{f} B \xrightarrow{g} A$ are arrows in an O-category K such that $g \circ f = Id_A$ and $f \circ g \sqsubseteq Id_B$, then we say that $\langle f,g \rangle$ is a *projection pair* of K.

The category having the same objects as K, but having as arrows the projection pairs of K (with the obvious definition of composition) is denoted KPR. If p is any projection pair, we denote by p^L the left part (the *embedding*) and by p^R the right part (the *projection*) of p. Similarly for chains, cones, etc.: thus if Δ is an ω-chain in KPR, Δ^L will be the ω-chain in K obtained by dropping the second components of the arrows of Δ. It will occasionally be useful to consider the category KP having the same objects as K, in which an arrow from A to B is just a pair $\langle f: A \to B, g: B \to A \rangle$ of K (composition as in KPR). KPR is of course a subcategory of KP.

DEFINITION 2. An object A of a category L is *finite* in L provided that, for any ω-chain $\Delta = \langle A_n, f_n \rangle_{n\,\omega}$ in L with colimit $\mu: \Delta \to V$, the following holds: for any arrow $v: A \to V$ and for any sufficiently large n, there is a unique arrow $u: A \to A_n$ such that $v = \mu_n \bullet u$. We say that L is *algebroidal* (= *algebraic* in [24]) provided (1) L has as initial object and at most countably many finite objects, (2) every object of L is a colimit of an ω-chain of finite objects, and (3) every ω-chain of finite objects has a colimit in L.

We shall be interested in O-categories K for which KPR is algebroidal. "Finiteness" of objects should always be understood relative to KPR. Notation: KPR_O (resp. K_O) is the full (O-) subcategory of KPR (K) with objects restricted to be finite. The following (new) definition is the cornerstone of this paper:

DEFINITION 3. An O-category K is *admissible* provided

(i) KPR has at most countably many finite objects.

(ii) If A, B are finite in KPR then Hom(A,B) is a finite set.

(iii) For every object A of K there is an ω-chain $\Delta = A_0 \to A_1 \to \ldots$ in KPR_O and a cone $\mu\colon \Delta \to A$ in KPR such that $\bigsqcup_i \mu_i^L \circ \mu_i^R = Id_A$. - A cone μ satisfying this condition will be called an *approximating* cone for A; we will also say that μ (or sometimes just A) is an O-*limit* of Δ.

(iv) Every ω-chain in KPR_O has an O-limit.

The principal definitions and results of this paper concern admissible categories. However, in order to capture more fully the idea of a category of *domains*, in which Scott's D_∞ constructions can be carried out, we need to specialize as follows:

DEFINITION 4. A *domain-category* is an admissible category K in which:

(i) Every Hom(A,B) has a least element $\perp_{A,B}$;

(ii) There is an object E such that $Id_E = \perp_{E,E}$;

(iii) For any arrow $f\colon A \to B$ and object C,

$$\perp_{B,C} \circ f = \perp_{A,C} .$$

An arrow $f\colon A \to B$ is called *strict* if $f \circ \perp_{E,A} = \perp_{E,B}$. The domain-category K is *strict* if every arrow K is strict.

The clauses of Definition 4 can be seen as extending the defining properties of admissible categories from ω-chains and sequences to finite (in particular, empty) chains and sequences . Thus, clause (i) says that in each hom-set the empty sequences has a lub; (ii) says that the empty chain in KPR_O has the O-limit E; (iii) gives one-solid distributivity ("continuity") over the empty sequence, while in the strict category we have full distributivity (it is easy to see that if f is strict we have $f \circ \perp_{C,A} = \perp_{C,B}$ for all C).

All the usual categories of algebraic domains with continuous functions are domain-categories (we are excluding the more general "continuous" domains; see Sec. 5), provided that isomorphic finite domains are identified. Cutting down to the strict functions (in the usual sense) gives a strict domain-category. A simple example of a strict domain-category is obtained by taking as objects the sets of natural numbers and as arrows the partial functions, the ordering of hom-sets being the usual inclusion of partial functions.

FACT. If K is a domain-category, then: E is terminal in K; E is initial in KPR; every embedding and projection is strict; if K is strict then E is initial in K; finally, the category KS obtained from K by cutting down to the strict arrows is a strict domain-category, and KSPR = KPR.

THEOREM 1. *Let* K *be an admissible* O*-category, and* $\mu\colon \Delta \to A$ *an approximating cone for*

A *in* K. *Then* μ *is colimiting in* KPR *and* μ^L *is colimiting in* K.

For the proof, see [26] Lemma 5.

COROLLARY. *If* K *is admissible and* KPR *has an initial object then* KPR *is algebroidal.*

The following result is relevant to the computability of the $\to$-functor (of type $K^{Op} \times K \to K$):

LEMMA 1. (i) K *is admissible iff* K^{Op} *is admissible.* (ii) *The product of two admissible* O*-categories is admissble.*

PROOF. (sketch). (i) Define a (1,1)-correspondence $X \to \bar{X}$ between entities (objects, arrows, w-chains,...) of KPR and entities of K^{Op}-PR by:

$$A \mapsto \bar{A} \qquad (\text{i.e.} \bar{\bar{A}} = A)$$

$$\langle f,g\rangle\colon A \to B \mapsto \langle g,f\rangle\colon A \to B$$

$$A_0 \overset{P0}{\to} A_1 \overset{P1}{\to} \ldots \mapsto A_0 \overset{\bar{P}0}{\to} A_1 \overset{\bar{P}1}{\to} \ldots$$

and similarly for cones. Using this correspondence, verify that: A is finite in KPR iff A ($=\bar{\bar{A}}$) is finite in K^{Op}-PR; and $\mu\colon \Delta \to A$ is an approximating cone in KPR iff $\bar{\mu}\colon \bar{\Delta} \to A$ is an approximating cone in K^{Op}-PR. The result follows at once.
(ii) Routine verification.

Projection pairs give us our notion of "approximation" between objects. We need also a notion of approximation for morphisms:

DEFINITION 5. Let K be an O-category. Then the *arrow-category* of K, K_{arr}, has as objects the arrows of K and as arrows from f to g (where $f\colon A \to A'$, $g\colon B \to B'$ in K) the pairs $\langle p\colon A \to A', q\colon B \to B'\rangle$ ($p,q \in$ KPR) such that

$$f \sqsubseteq q^R \circ g \circ p^L.$$

Assume now that K is admissible. Let $\Delta = A_0 \overset{f_0}{\to} A_1 \to \ldots$, $\Gamma = B_0 \overset{g_0}{\to} B_1 \to \ldots$ be ω-chains in KPR_0 with limits $\mu\colon \Delta \to A$, $\nu\colon \Gamma \to B$, and assume that arrows $h_i\colon A_i \to B_i$ satisfy

$$h_i \sqsubseteq g_i^R \circ h_{i+1} \circ f_i^L \qquad (i = 0,1,\ldots)$$

(so that we have an ω-chain H in K_{Arr}). Then a O-*limit* of H is a cone $\xi\colon H \to \bar{h}_i$ in K_{Arr}, where $\xi_i = \langle \mu_i, \nu_i\rangle$, $\mu\colon \Delta \to A$ and $\nu\colon \Gamma \to B$ are O-limits, and $\bar{h}_i\colon A \to B$ is given by $\bar{h}_i = \nu_i^L h_i \mu_i^R$ (we also say that ξ is an *approximating* cone for $h = \bigsqcup_i \bar{h}_i$). We will

often omit the prefix O- of O-limit.

It is easily checked that $\langle \bar{h}_i \rangle$ is increasing, so that this definition is sound, and a limit for an ω-chain H as described always exists; furthermore, limits are unique up to isomorphism of cones, so that we may as well write *the* limit. By an abuse we will also write, for example, "$\bar{h}_i$ is a limit of the h_i", the connecting arrows and cone being understood.

Various elementary properties of the category K_{Arr} can at once be established: for example, the "limit" just defined is actually the colimit in K_{Arr}, and K_{Arr} has as finite objects exactly the finite arrows of K (that is, the arrows $f: A \to B$, A, B finite in KPR). But we shall not need these in the present work.

THEOREM 2. *In an admissible O-category* K, *every arrow can be expressed as the limit of some ω-chain of finite arrows.*

PROOF. Let $h: A \to B$ be any arrow in K, and let $\mu: \Delta \to A$, $\nu: \Gamma \to B$ be limits, where $\Delta = A_0 \overset{f_0}{\to} A_1 \to \ldots$, $\Gamma = B_0 \overset{g_0}{\to} B_1 \to \ldots$ are chains in K_0. Put $h_i = g_i^R \circ h_{i+1} \circ f_i^L$. Further

$$\bigsqcup_i \nu_i^L \circ h_i \circ \mu_i^R = \bigsqcup_i \nu_i^L \circ \nu_i^R \circ h \circ \mu_i^L \circ \mu_i^R = \left(\bigsqcup_i \nu_i^L \circ \nu_i^R\right) \circ h \circ \left(\bigsqcup_i \mu_i^L \circ \mu_i^R\right) = h.$$

So h is the limit of $\langle h_i \rangle_i$.

Notice that the sequence $\langle h_i \rangle_i$ constructed here has the strong property that $h_i = g_i^R \circ h_{i+1} \circ f_i^L$, where only $\sqsubseteq$ is required for an "approximating" cone in K_{Arr}. Intuitively, the difference is that the strong sequence $\langle h_i \rangle_i$ gives the *best* approximation to h that is available at stage i. It might be asked why we do not work with the strong sequences which have this (apparently) attractive property. The main reason has to do with effectiveness: even when h is computable, in an appropriate sense, we cannot (in general) *effectively* list the h_i (more precisely: list explicit indices for the h_i, see Sec. 3.2).

An important point in our investigation is that the operations / relations over objects and arrows which we want to discuss can be all expressed in terms of operations / relations over the finite entities in the approximating cones for those objects and arrows. Example:

LEMMA 2. *Let* $\langle h: A \to B, p: B \to A \rangle$ *be a pair of arrows in the admissible category* K. *Let* $\mu: \Delta \to A$, $\nu: \Gamma \to B$ *be approximating cones for* A, B, *where* $\Delta = A_0 \overset{f_0}{\to} A_1 \to \ldots$, $\Gamma = B_0 \overset{g_0}{\to} B_1 \to \ldots$, *and let* $\langle h_i: A_i \to B_i \rangle_i$, $\langle p_i: B_1 \to A_i \rangle_i$ *be ω-chains of arrows having* h, p *as limits (via* μ, ν*). Then* $\langle h,p \rangle$ *is a projection pair iff*

(i) *for all* k, *and for all sufficiently great* i ($\geq k$), $f_{ik} \circ p_i \circ h_i \circ f_{ki} = Id_{A_k}$, *and*

(ii) *for all* k, $h_k \circ p_k \sqsubseteq Id_{A_k}$.

PROOF. Only if:

(i) $$Id_{A_k} = \mu_k^R \circ p \circ h \circ \mu_k^L = \mu_k^R \circ (\bigsqcup \mu_i^L \circ p_i \circ \nu_i^R) \circ (\bigsqcup \nu_i^L \circ h_i \circ \mu_i^R) \circ \mu_k^L$$

$$= \bigsqcup \mu_k^R \circ \mu_i^L \circ p_i \circ h_i \circ \mu_i^R \circ \mu_k^L = \bigsqcup_{i \geq k} f_{ik} \circ p_i \circ h_i \circ f_{ki} \ .$$

Since there are only finitely many terms in this (increasing) sequence (Def. 3, (ii)), all of them from some point on must be Id_{A_k}.

(ii) $$h_k \circ p_k \sqsubseteq \nu_k^R \circ h \circ \mu_k^L \circ \mu_k^R \circ p \circ \nu_k^L \sqsubseteq \nu_k^R \circ h \circ p \circ \nu_k^L \sqsubseteq Id_{B_k} \ .$$

If: Straightforward.

An alternative formulation is: $\langle h,p \rangle$ is a projection pair iff, for all k and for all sufficiently great i ($\geq k$), $\langle h_i \circ f_{ki}, f_{ik} \circ p_i \rangle$ is a projection pair. From this in turn we can get a simple characterization of *isomorphism* pairs h, p.

Simple characterizations of $f \sqsubseteq f'$ and of $f = f'$, in the same style, may be obtained. A particularly important example is *composition*: this is just given by pointwise composition in the approximating chains. That is, if f, g are given as limits of $\langle f_i : A_i \to B_i \rangle$, $\langle g : B_i \to C_i \rangle$ then $g \circ f$ is the limit of $\langle g_i \circ f_i : A_i \to C_i \rangle$. Note that this does not work if "strong" approximating sequences are used: the strong commuting property does not carry over to the composed diagram. This technical difficulty - which becomes quite troublesome when we come to consider effective sequences - is another reason for preferring the weak sequences.

3.1. Effectiveness

DEFINITION 5. An *effective basis* for an admissible O-category K consists of an enumeration $O_0, O_1, \ldots$ of the finite objects (of KPR) and an enumeration $m_0, m_1, \ldots$ of the finite arrows of K, satisfying the following:

(i) The predicate $O_m = O_n$ is recursive in indices.

(ii) The predicate "m_n is an identity" is recursive.

(iii) There are recursive functions D, C such that

$$O_{D(n)} = dom(m_n) \quad \text{and} \quad O_{C(n)} = cod(m_n) \ .$$

(iv) There is a partial recursive function Comp such that $m_k \circ m_n = m_{Comp(k,n)}$, where Comp is defined iff the composition exists.

(v) The predicate $m_k \sqsubseteq m_n$ is recursive in indices (false if m_k, m_n are not in the same hom-set).

Informally: Equality of (finite) objects is decidable, as are the orderings of the hom-sets; whether an arrow is an identity is decidable; and the domain, codomain and composition of arrows are computable (in indices).

In the following work we make heavy use of recursive sequences of objects/arrows in an effective basis, given as, say, $\langle O_{e(i)} \rangle_i$, $\langle m_{f(i)} \rangle_i$ (e, f recursive functions). To spare the notation we will often just write the index e(i) (f(i)) in place of $O_{e(i)}$ ($m_{f(i)}$).

DEFINITION 6. An *effectively given* O-category is an admissible category K together with an effective basis of K (Notation K^E). An *effective finitary projection sequence* (or *basic sequence*) in K is an effective ω-chain in KPR_O, that is, an ω-chain in KPR_O of the form $e(0) \xrightarrow{f(0)} e(1) \to \ldots$, e, f recursive. An *effectively given* object of K is an object A of K together with an approximating cone $\mu\colon \Delta \to A$ for A, where Δ is basic. An *effectively given* (or *computable*) arrow in K is an arrow h together with an approximating cone $\Phi\colon \quad \to h$ for h, where Γ is basic (i.e. Γ is of the form

$$
\begin{array}{ccccc}
 & e(0) & \xrightarrow{f(0)} & e(1) & \ldots \\
(1) \qquad g(0) & \downarrow & & g(1) & \downarrow \\
 & e'(0) & \xrightarrow{f'(0)} & e'(1) & \ldots
\end{array}
$$

e, f, e', f', g recursive).

This definition is in the spirit of Kanda and Park [13], where it is shown that an effectively given basis of a domain must be understood to include a particular enumeration of the basis, and an effectively given domain must be taken as a domain D *together with* an effective basis for D (not just: for which *there exists* some effective basis).

We still have to explain the exact relation between our categorical formulation of effectiveness and the familiar formulations in terms of domains. To be specific, let us take K = ACPO (bounded-complete algebraic cpo's, continuous maps). We will show in the next subsection that, relative to suitable indexings of the finite objects of ACPO-PR (i.e. the finite domains) and of the finite arrows of ACPO, the effectiveness of objects and arrows given by Definition 6 coincides with that studied in [5], [23].

3.2. Effectively given domains

We are going to require that the finite domains (in ACPO) and maps between finite domains be enumerated by a system of *explicit* indices. An explicit index for a finite structure is an index from which we can recover the whole structure in a "time" (number of steps) bounded in advance. The term "explicit" is borrowed from Myhill and Shepherdson [18]; it corresponds to Rogers' "canonical" [20]. Explicit indices for domains might be constructed, say, as follows: Suppose that c is an explicit index of a finite set C, and that r is an explicit index of a binary relation

R on C which gives the structure of a finite domain; then <c,r> is an *explicit index* of this domain. This gives only a partial indexing, of course; we suppose it made total in some obvious way. Explicit indices for the maps are constructed similarly. It is obvious that enumerations of the finite objects and arrows of ACPO, using explicit indices, yield an effective basis for ACPO (Definition 5).

We claim that, given an effective basis built in this way (a *standard* effective basis, we shall say), the notions of effectively given object and arrow obtained from Definition 6 agree with the usual notions of effectively given domain and computable functions. (For reasons of space, we omit all details here; see [27].)

4. COMPUTABLE FUNCTORS

Corresponding to the notion of a graph of a function we introduce *frames* of a functor. A frame of a locally continuous functor $F: K \to K'$ (K, K' admissible) is an O-functor $F_0: K_0 \to \bar{K}'_0$ such that for each finite object A (finite arrow f) of K, $F_0(A)$ ($F_0(=f)$) is an approximating chain for F(A) (F(f)). In case K, K' are effectively given, we have the notion of a *recursive* frame F_0: the object and morphism maps of F_0 are given by recursive index maps. Of course, functors do not have unique frames; but we shall find that computability of functors can be defined in terms of frames in much the same way as computability of functions in terms of graphs.

DEFINITION 7. Let K, K' be effectively given O-categories. A *computable* functor from K to K' is a locally continuous functor $F: K \to K'$ which has a recursive frame.

THEOREM 3. *A computable functor* $F: K \to K'$ *preserves computability of objects and morphisms. More precisely, if* D *is effectively given then a basic sequence can be given for* F(D)*, and similarly for morphisms. Moreover, the object and morphism maps so defined are recursive in indices.*

In practice the basic functors used in semantics ($+, \times, \to, p, \ldots$) always map finite domains/functions to finite domain/functions, and the computability of these functors is immediate. Theorem 3 assures us that application of these functors does not take us outside the realm of the computable. More interesting is the question of the computability of initial fixpoints (algebras) of such functors. This can, indeed, be ensured by restricting to domain-categories which are effectively given (as admissible categories). Probably the easiest way to handle this topic is to begin by showing that, for effectively given categories, our computable functors are essentially the same as Kanda's *effective* functors ([11], [12]) - something which follows easily from Theorem 3 - and then borrow the results of [12]. It is intended to publish the details on another occasion.

In connexion with Kanda's work, let us mention, finally, his exhaustive study (in [11]) of effectiveness in Plotkin's category SFP. Kanda finds that the definition of effectiveness of an SFP object in terms of suitable conditions on the basis of

finite elements is very troublesome to work with, especially in connexion with the function space construction. Purely on grounds of simplicity of the resulting theory, he is led to a definition in terms of colimits of ω-chains of finite domains, in close agreement with our abstract Definition 6.

5. MISCELLANEOUS

(1) O and ω. A striking feature of the category-theoretic treatment of data types ([16], [26]) is the interplay between "O-notions", deriving from the ordering of hom-sets, and "ω-notions", having to do with colimits of ω-chains. O-notions are appropriate to categories of (domains and) continuous functions, ω-notions to categories of projection pairs. Both types of notion seem to be necessary to get a satisfactory theory of data types, and so it is perhaps useful to have an independent set of definitions of effectiveness for the ω-notions. We say "perhaps" since in fact it is nearly always most convenient to establish ω-properties via corresponding O-properties (see [26] for examples), so that effectiveness of the O-properties may be sufficient. In any case it is easy to give direct definitions of *effectively given* ω-category, *computable* ω-functor, and so on. For the effectively given ω-category, for example, we just incorporate some effectiveness conditions into the definition of an algebroidal category; this is similar to, but simpler than, Definitions 2, 4 above.

(2) Adjoint functors. Much use of adjoint functor theorems has been made in, for example, [8] and [15]. This is suspect from our point of view, since the "constructions" given in the usual proofs of these theorems are highly non-effective. One could look for restricted versions of these theorems that could readily be made effective. For example we note that, often, the functors for which we wish to construct adjoints preserve finiteness (of objects and arrows); they may be regarded as functors between "bases" K_0 of admissible categories K. Let us then restrict attention to countable O-categories in which no two distinct objects are isomorphic and in which all hom-sets are finite. Let us say that such a category is *finitely powered* if every object has only finitely many subobjects. For these categories we have the following version of the Special Adjoint Functor Theorem of [9]:

THEOREM 4. *Suppose that* K *is finitely powered, finitely complete, and has a coseparator. Then an* O*-functor* $G: K \to X$ *has a left adjoint iff* G *preserves finite limits.*

One can immediately write down an *effective* version of this theorem (relative to suitable indexings of K, X).

We doubt that such theorems are really very useful. It seems that it is usually just as easy to construct the adjoint for a "basis" functor G directly, as to appeal to general theorems. What is, perhaps, more useful is to extend adjointness from functors between "bases" K_0, L_0 to functors between the completed (admissible) cateogries K, L:

THEOREM 5. *Let* F: K→L *be a computable functor which preserves finiteness, so that* F *has in effect a recursive frame* F_0: $K_0 \to L_0$. *Suppose that* F_0 *has a recursive left (right) adjoint* G_0: $L_0 \to K_0$. *Then* F *has a computable left (right) adjoint* G: K→L *with* G_0 *as frame.*

Our construction of the power-domains in [24] can - by taking into account [8] - be seen to fit this pattern (it would have done so still more closely had we followed the "first variant" mentioned in Sec. 4 of [24]).

(3) The definitions we have given above are appropriate only to categories of "algebraic" domains. An abstract treatment capable of handling continuous domains can be developed using a construction of Freyd [7], p. 51. The construction yields, for any category K, a canonical extension of K in which all idempotents split. The extended category might be called the *category of retracts* over K ([25], Ch. 3, Sec. 2.2); given a category of algebraic domains such as ACPO it yields in effect the corresponding category of continuous domains (here, CCPO). However, such a development seems more trouble than it is worth, especially it is still not clear whether non-algebraic domains are really needed in computer science.

(4) A weakness of our notion of effectively given category is that it is not closed under the formation of functor categories. (Functor categories are useful in the theory of data types: [16] Sec. 4.) The situation is analogous to that obtaining in the theory of domains. As is well-known, the class of countably-based algebraic cpo's is not closed under function space formation; we have to restrict at least to SFP to achieve closure. The same solution is, we believe, appropriate here: to restrict to categories which may be considered as limits of finite categories (so eliminating the "lapse" mentioned in Section 1). This is currently under investigation.

REFERENCES

1. Berry, G., Modeles completement adequats et stables des lambda-calculus types, These, Universite Paris VII, 1979.
2. Bishop, E., *Foundations of Constructive Analysis*, McGraw-Hill (N.Y.), 1967.
3. Constable, R., Constructive mathematics and automatic program writers, IFIP (1972), 229-233.
4. Curien, P., Algorithmes sequentiels sur structures de donnees concretes, These de 3e cycles, Universite Paris VII, 1979.
5. Egli, H. & R. Constable, Computability concepts for programming language semantics, *Theor. Comp. Sci.* 2 (1976).
6. Ehrich, H.-D. & V. Lohberger, Parametric specification of abstract data types, parametric substitution, and graph replacements, Proc. of workshop on "Graphentheoretische Konzepte in der Informatik", ed. - J. Mühlbacher, Hanser-Verl., München, 1977.

7. Freyd, P., *Abelian Categories*, Harper and Row, 1964.
8. Hennessy, M. & G. Plotkin, Full abstraction for a simple parallel programming language, MFCS '79, LNCS 74, 1979.
9. Herrlich, H. & D. Strecker, *Category Theory*, Allyn and Bacon, 1978.
10. Kahn, G. & G. Plotkin, Domaines concrets, Rapport No. 336, IRIA Laboria, 1978.
11. Kanda, A., Thesis, University of Warwick, 1979.
12. Kanda, A., Fully effective solutions of recursive domain equations, MFCS '79, LNCS 74, 1979.
13. Kanda, A. & D. Park, When are two effectively given domains identical?, *Proc. 4th GI Conf. in T.C.S.*, Aachen, LNCS 67, 1979.
14. Lehmann, D., Categories for fixpoint semantics, Theory of Computation Report No. 15, University of Warwick, 1976.
15. Lehmann, D., On the algebra of order, Mathematics Institute, Hebrew University, Jerusalem.
16. Lehmann, D. & M. Smyth, The algebraic specification of data types; a synthetic approach, Report No. 119, Dept. of Computer Studies, University of Leeds, 1978 (also to appear in Math. Syst. Theory).
17. Martin Löf, P., Constructive mathematics and computer programming, 6th Int. Congress for Logic, Methodology and Phil. of Science, Hannover, 1979.
18. Myhill, J. & J. Shepherdson, Effective operations on partial recursive functions, *Zeitschr. f. math. Logik u. Grundl. d. Math.*, 1956.
19. Plotkin, G., A power-domain construction, *SIAM J. Comput.* 5 (1976), 452-487.
20. Rogers, H., *Theory of Recursive Functions and Effective Computability*, McGraw-Hill, 1967.
21. Scott, D., Continuous lattices, *Lecture Notes in Math.* 274, Springer, 1974.
22. Scott, D., Data types as lattices, *SIAM J. Comput.* 5 (1976), 522-587.
23. Smyth, M., Effectively given domains, *Theor. Comp. Sci.* 5 (1977), 257-274.
24. Smyth, M., Power Domains, *J. Comp. Syst. Sci.* 16, (1978), 23-26.
25. Smyth, M., D. Phil Thesis, University of Oxford, 1978.
26. Smyth, M. & G. Plotkin, Category-theoretic solution of recursive domain equations, D.A.I. Report 60, University of Edinburgh, 1978.
27. Smyth, M., Computability in Categories, Theory of Computation Report, University of Warwick, 1979.
28 Tang, A., Recursion theory and descriptive set theory in effectively given T_0 spaces, Ph.D. Thesis, Princeton University (1974).
29. Bergstra, J. & J. Tucker, Algebraic specifications of computable and semi-computable data structures, Dept. of Computer Science Report 1W115, Math. Cent., Amsterdam, 1979.
30. Wand, M., Fixed-point constructions in order-enriched categories, TR 23, Computer Science Dept., Indiana University, 1975.