TOWARDS A THEORY OF ABSTRACT DATA TYPES:

A DISCUSSION ON PROBLEMS AND TOOLS (°)

A. Bertoni, G. Mauri, P. Miglioli (°°)

Abstract: This paper aims to show that, in order to capture a quite relevant feature such as the recursiveness of abstract data types, Model Theory works better than Category Theory.

First, various categorial notions such as "initiality", "finality", "monoinitiality", "epifinality", "weak monoinitiality" and "weak epifinality" are analyzed, from a model theoretic point of view, as regards the "abstractness" and the "cardinality" of the models they determine. In particular, countability is seen as a necessary condition to get recursive data types, and it is shown that only "initiality", "monoinitiality", "epifinality" and "weak epifinality" allow to select countable models.

An extensive analysis is then devoted to the problem of the recursiveness of abstract data types: we provide a formal definition of recursiveness and show that it neither collapses, nor it is incompatible with the "abstractness" requirement. We also show that none of the above quoted categorial notions captures recursiveness.

Finally, we consider our own definition of abstract data type, based on model-theoretic notions; we analyze this definition in the frame of the proposed formalization of recursiveness, and illustrate the sense according to which it captures recursiveness.

Keywords: abstractness, countability, homomorphism, isomorphic embedding, recursiveness.

(°) This research has been supported by the Grant n.79 00701 02 of C.N.R.

(°°) Istituto di Cibernetica, Università di Milano
Via Viotti 5, 20133 MILANO, Italy

1. INTRODUCTION

While the advantages of data abstractions in programming methodology are universally acknowledged, a general agreement on a comprehensive theory of data types has not yet been achieved, and every author, having different motivations and aims, emphasizes different aspects of data types and takes the approach that better captures such aspects. In particular, while everyone agrees on the power of axiomatic definitions as a tool for formal specification of abstract data types, different opinions arise about the restrictions to be imposed on the form of axioms, in order to be sure that we can specify exactly those structures which can be reasonably considered (for Computer Science) data types.

In a previous paper [1], we gave a complete formalization of our own approach, based on model theoretic techniques. A further development of this approach enforced our opinion that purely algebraic techniques are not sufficiently powerful in order to specify abstract data types. In this paper, we will discuss a few critical points about the notion of data type, and will attempt a comparative analysis of the power of algebraic and model theoretic tools in order to overcome them.

The first critical point concerns the notion of data type itself, and will be briefly discussed in the next section; a formal setting will be found in [1].

The meaning of "abstractness" is discussed in the third section, where a comparison of a number of different "algebraic" concepts which have been (or might be) used to capture abstractness is given. The comparison of the algebraic concepts is completed in the fourth section, where we look at them from a general model theoretic point of view and give a number of results concerning the cardinality of the models corresponding to the various concepts.

Finally, a section is devoted to the fundamental problem of the recursiveness of data types: here, it is proved that our approach is the only, among the ones considered in the paper, which captures "recursiveness".

In the following, the reader will be assumed to be acquainted with the basic definitions of category theory [10], first order logic and model theory [4,7]. In particular, we assume as known: the notion of homomorphism, both in the usual algebraic context, where only operations are taken into account, and in a general model theoretic context (see [4] pag.70), where relations are considered, too; the notions of monomorphism and epimorphism [10]; the notion of signature of a first order language ([4] pag.18), extended to the heterogeneous case [5]; the notions of first order lan-

guage , of sentence , of first order theory (generalized to the heterogeneous case, as in [1]), of model of a first order theory (with the usual notations $\mathcal{M} \models \mathcal{T}$ and $\mathcal{M} \models H$ denoting respectively that $\mathcal{M}$ is a model of the theory $\mathcal{T}$ and that the wff H holds in the model $\mathcal{M}$) [4,7]; the notion of diagram of a first order model ([4] pag. 68; the diagram of $\mathcal{M}$ will be denoted by $D(\mathcal{M})$ or by $D_C(\mathcal{M})$ when we want to put in evidence that it is built up by introducing the set of constants C); the notion of provability in a first order theory (as usual, $\mathcal{T} \vdash H$ will mean that the wff H is provable in $\mathcal{T}$). Moreover, we assume as known the main theorems of model theory, in particular the completeness and the compactness theorems and the downward Loewenheim-Skolem theorem.

Of course, for abstract data specification the considered first order language is required to be countable; the results of section 4 explicitly depend on this requirement. In Section 5 we also require that a theory be axiomatizable [7], i.e. that the set of all its axioms be recursively enumerable. We recall the following result due to Craig: a theory is axiomatizable iff it is equivalent to a theory with a recursive set of axioms.

2. DATA AND DATA TYPES

A nearly general agreement has been reached in considering a data type as characterized by objects and operations [12]; this leads to Algebra as the natural frame to formalize such a concept [5,6,15]; more precisely, the notion of data type is formalized by the notion of heterogeneous equational algebra, specified by:

- a finite set of sorts;
- a finite set of operation symbols, with a finite arity;
- a finite set of equations (universally quantified equalities between terms).

In this approach, the primitive concept is that of data type rather than of datum, and the restrictions on the form of the axioms are imposed as an "a priori". On the contrary, a model theoretic approach, as developed in [1], allows to first formalize the concept of datum, which is exactly captured by the notion of model theoretic invariant [8], i.e. of a formula Δ provable in a first order theory $\mathcal{T}$ and such that every model $\mathcal{M}$ of $\mathcal{T}$ contains a unique element satisfying Δ . From this definition, the notion of data type and a characterization of theories which specify data types follow in a natural way (a sketch of the characterization is given in Sec.5).

A second aspect under which the algebraic approach seems to us to be inadequate refers to the specification of data types with relations together with operations:

an algebraic specification of relations is possible only in the form of functions on the type `BOOL`, so imposing undue restrictions.

3. THE PROBLEM OF ABSTRACTNESS

The second relevant problem to be stressed concerns the "abstractness" requirement for data types and the tools which better capture such an abstractness.

Informally, abstractness can be intended as independence of any implementation of data and operations; in a theoretic frame, this can be settled as the problem of identifying a model unique up to isomorphisms. It is well known that first order theories (save trivial cases) cannot do this: indeed, if a first order theory has an infinite model, then it has a model of every greater cardinality. Thus, the uniqueness (up to isomorphisms) can be guaranteed only by using implicitly second order notions.

Category theory [10] has provided notions which seems to be adequate to overcome the above problem: we mean notions such as initiality [5] and finality [16] or monointiality and epifinality (see below); furthermore, weaker concepts such as terminality [3] have been introduced: terminality selects an unique model only in the restricted category of finitely generated models.

DEF.3.1. - Let τ a first order theory, and $\mathcal{C}_\tau$ the category of models of τ ; a model $M \in \mathcal{C}_\tau$ is said to be:

- a) initial (final) in $\mathcal{C}_\tau$ iff for every $M' \in \mathcal{C}_\tau$ there is a unique morphism $f: M \rightarrow M'$ ($f: M' \rightarrow M$);
- b) monointial (weakly monointial) iff for every $M' \in \mathcal{C}_\tau$ there is a unique monomorphism (respectively, at least one monomorphism) $m: M \rightarrow M'$;
- c) epifinal (weakly epifinal or terminal) iff for every $M' \in \mathcal{C}_\tau$ there is a unique (resp., at least one) epimorphism $e: M' \rightarrow M$.

We remark that the present notion of monointiality is quite different from the one considered in [1], which in [2] and in the present paper is called isointiality, since monomorphisms are not required to be injective.

The following facts, some of which have been stated in [2], can be proved:

- a) An initial, final, monointial or epifinal model, if it exists in $\mathcal{C}_\tau$, is unique up to isomorphisms.
- b) Initiality, finality, monointiality and epifinality are independent notions, i.e. the presence in $\mathcal{C}_\tau$ of any model corresponding to one of these notions does not imply the presence of a model corresponding to another such notion.
- c) If an initial and a monointial model coexist in $\mathcal{C}_\tau$, then they are isomorphic;

the same holds for final and epifinal models.

- d) there are categories of models $\mathcal{C}_\tau$ where non isomorphic initial and final (monoinitial and epifinal) models coexist.
- e) if in $\mathcal{C}_\tau$ a monoinitial and a weakly monoinitial (an epifinal and a weakly epifinal) model coexist, then they are isomorphic.
- f) there are categories of models where infinitely many non isomorphic weakly monoinitial models coexist together with infinitely many non isomorphic terminal models.
- g) if in $\mathcal{C}_\tau$ there are two non isomorphic weakly monoinitial (terminal) models $\mathcal{M}$ and $\mathcal{M}'$, then they are necessarily infinite.

4. THE PROBLEM OF CARDINALITY

Abstractness is not the only requirement to be satisfied by data types. Quoting Standish [14], "there are additional properties required of computer data that are not required of representations in general. For example, in mathematics we can use uncountable, continuous representations, but in digital computers our representations must be countable and discrete". Hence, it is needed to analyze the models corresponding to the notions introduced in Sec.3 from the point of view of their cardinality. Indeed, while in an "algebraic theory of data" an initial model, being defined as a quotient of a word algebra, is trivially countable, and a final model, by the absence of negative axioms, collapses into the trivial one-element model, in a "model-theoretic theory of data" nothing is immediately clear.

Our first two results show, perhaps surprisingly, that in a model theoretic frame it is just finality that does not imply countability. Such a contrast between two concepts that in category theory are seen as dual, i.e. interchangeable in some sense, should show that specific features about data types can be captured only at a different level of generality, i.e. in model theory rather than in category theory.

From now on, τ will be a countable first order theory, and $\mathcal{C}_\tau$ the category of models of τ .

Th.4.1 - If a model $\mathcal{M}$ is initial or monoinitial in $\mathcal{C}_\tau$, then it is countable.

The proof is an easy application of the downward Loewenheim-Skolem theorem.

Th.4.2 - Let $\mathcal{F}$ be a final model in $\mathcal{C}_\tau$; then $\mathcal{F}$ has at most the cardinality of continuum. On the other hand, there are finitely axiomatizable theories with universal axioms whose final model has exactly the cardinality of continuum.

We omit the proof of the first part of Th.4.2, which involves the downward Loewenheim-Skolem theorem and a compactness argument. For the second part, we provide

the following example.

Example: STACKS WITH A LEXICOGRAPHICAL ORDERING (S.O.L.)

Sorts: $A(\text{set}), S(\text{stacks})$

Operations: $0 \in A, 1 \in A, \lambda \in S, \therefore AxS \rightarrow S, t:S \rightarrow A, p:S \rightarrow S$

Relations: $\leq \subseteq S \times S$

Axioms: A contains only 0 and 1 (i.e., $x_A = 0 \vee x_A = 1, 0 \neq 1$)

., t, p are the usual PUSH, TOP and POP (i.e., $\forall x(x \neq \lambda \rightarrow x = t(x).p(x)),$
 $p(\lambda) = \lambda, t(\lambda) = \lambda$, etc.)

$\leq$ is a total ordering relation (i.e., $x \leq x, x \leq y \wedge y \leq x \rightarrow y = x, x \leq y \wedge y \leq z \rightarrow x \leq z, x \leq y \vee y \leq x$), which is, moreover, a lexicographical ordering (i.e., $y \neq \lambda \rightarrow (p(x) > p(y) \wedge t(x) = t(y) \rightarrow x > y), 1.y > 0.x, 0.x > \lambda, 1.x > \lambda$, where $x > y$ means $x \not\leq y$).

Let A^* be the set of all (finite) strings generated by $A = \{0, 1\}$ and let A^ω be the set of all infinite strings generated by A ; then the model:

$\mathcal{A} = \langle A, A^* \cup A^\omega, ., p, t, \leq \rangle$, where: $\sigma.x = \sigma x, p(\sigma x) = x, t(\sigma x) = \sigma, x > x'$ iff $x = sly$
 and $x' = sy'$ for some standard string s

turns out to be a final model with the power of continuum.

The next theorem concerns the cardinality of terminal models; its proof uses the downward Löwenheim Skolem in its strongest form (where the notion of "elementary submodel of a model, see [4,7], is required) and involves a compactness argument.
 Th.4.3 - Every terminal model of a theory $\mathcal{T}$ with a countable set of wff is countable.
 Corollary - The epifinal model of $\mathcal{T}$ (if it exists) is countable.

Quite surprisingly, for weakly monoinitial models we can prove the following theorem, which introduces a second asymmetry with respect to a purely categorial point of view:

Th.4.4 - There is a theory with a set of countable wff such that the category $\mathcal{C}_{\mathcal{T}}$ contains a weakly monoinitial model with the power of continuum.

According to Th.4.4, the second asymmetry is, so to say, opposite to the previous one: for, initiality involves "starting homomorphisms", terminality involves "arriving epimorphisms".

5. THE PROBLEM OF RECURSIVENESS

In our view, countability corresponds to a minimum of constructiveness to be required for abstract data specification. In this section we will take into account a much stronger condition, i.e., we will be faced with the following questions:

I) Is there a sense according to which an abstract data type can be said to be recursive?

II) Assuming that one can reasonably define what is to be meant by "a recursive abstract data type", are there criteria to select such abstract data types?

In literature the above questions either have not been taken into account (e.g. [4]), or one implicitly assumes that an abstract data type needs to be recursive according to some "informal sense" [11]. We will try to provide a partial answer to both questions, by first proposing a particular definition of "recursive model of a first order theory", according to the lines outlined in [1] and [2]: in this sense, we propose our definition only as a starting point, a formal suggestion in order to carry out further investigations, simultaneously involving Recursiveness and Model Theory, according to the goals of Computer Science (for instance, complexity aspects should be taken into account).

An answer to the second question in terms of a typically model theoretic kind of homomorphisms, usually called "isomorphic embeddings" is proposed. The latter notion will allow us to introduce, as in [2], the concept of "isoinitial model of a first order theory" as a recursive abstract data type.

We will also explain that, as compared to the previously defined criteria based on the more general notion of "homomorphism", isoinitiality is the only defined criterion guaranteeing recursiveness. In this line, as a further illustration of what is involved in the recursiveness of the isoinitial models, we will see that a somehow weaker notion based on the isomorphic embeddings too, i.e. "primeness", does not capture recursiveness.

Here we require that a theory be axiomatizable, i.e., that the set of its axioms be recursively enumerable [4].

Coming to the first question, the main problem is that recursiveness seems to contrast with the "abstractness" of a data type. For, abstractness implies that a data type can be considered independently of any implementation of its objects and operations; recursiveness, on the other hand, is inherently connected with particular representations and is not preserved under isomorphisms. To better explain what we mean, we will consider two examples.

Example 1. Let $\mathcal{T}$ be a one sorted theory whose signature contains only the unary relation-symbol P and let, for every $n \geq 1$, H_n and K_n be respectively the wff $\exists x_1 \dots \exists x_n (x_1 \neq x_2 \wedge \dots \wedge x_1 \neq x_n \wedge \dots \wedge x_{n-1} \neq x_n \wedge P(x_1) \wedge \dots \wedge P(x_n))$ (there are at least n elements satis-

fying P) and the wff $\exists x_1 \dots \exists x_n (x_1 \neq x_2 \wedge \dots \wedge x_1 \neq x_n \wedge \dots \wedge x_{n-1} \neq x_n \wedge P(x_1) \wedge \dots \wedge P(x_n))$ (there are at least n elements satisfying $\neg P$): the axioms of $\mathcal{T}$ will be all formulas H_n and K_n , for $n=1, 2, \dots$ (clearly, $\mathcal{T}$ is axiomatizable).

Now, all models of $\mathcal{T}$ are infinite and all countable models are isomorphic to the structure $\mathcal{M} = \langle N, E \rangle$, where N is the set of the natural numbers and P is the set E of the even numbers. Of course, $\mathcal{M}$ is recursive, but the isomorphism class of $\mathcal{M}$ contains models $\mathcal{M}' = \langle N, E' \rangle$, where E' is any non recursive set of natural numbers: as we can pass from the recursive structure $\mathcal{M}$ to the non recursive $\mathcal{M}'$ with a non recursive isomorphism, likewise we can pass from the non recursive structure $\mathcal{M}'$ to the recursive $\mathcal{M}$ with a non recursive isomorphism.

Example 2. In the above example the non recursive isomorphism from $\mathcal{M}'$ to $\mathcal{M}$ is of no interest since, presumably, one directly takes into account $\mathcal{M}$ without passing through $\mathcal{M}'$; but there are more problematic situations. For instance, let $\mathcal{T}'$ be a one sorted theory whose signature contains only the binary relation-symbol $<$ and the unary relation-symbol Q : $<$ is axiomatized as an irreflexive linear ordering among infinite elements, in such a way that there is a least element (the "zero") and, for every x , there is a y which is the least element greater than x (y is the "successor" of x ; remark that neither "zero" nor "successor" are operations "declared" in the signature of $\mathcal{T}'$); the predicate Q is axiomatized as follows.

First of all, for every n one can write down a well formed formula $H_n(x)$ of $\mathcal{T}'$ saying that x is the n -th "successor" of "zero"; then, one chooses a set S of natural numbers which is simple in the usual sense of Recursiveness Theory (S is a coinfinite recursively enumerable set whose complementary does not contain infinite recursively enumerable subsets [16]) and axiomatizes Q with the axiom-schema: $\exists x (H_n(x) \wedge Q(x))$ for every $n \in S$, with the further axiom: $\forall x \exists y (x < y \wedge \neg Q(y))$ (clearly, $\mathcal{T}'$ is axiomatizable since, being S recursively enumerable, it can be specified by a recursively enumerable set of axioms).

Now, let $\mathcal{G} = \langle N, <, T \rangle$ be any structure consisting of the natural numbers with the usual irreflexive ordering, with T any coinfinite set containing the simple set S : if we interpret the predicate Q as T , we get a model of $\mathcal{T}'$ (more specifically, $\mathcal{G}$ turns out to be a "prime model" of $\mathcal{T}'$, in the sense specified below; since T is any coinfinite set containing S , $\mathcal{T}'$ has infinitely many non isomorphic prime models). We propose the following question: is there some structure $\mathcal{G}' = \langle N, <', T' \rangle$ which is isomorphic to $\mathcal{G}$ and is such that T' is recursive, while, at the same time, the non recursive isomorphism $i: \mathcal{G} \rightarrow \mathcal{G}'$ preserves the recursiveness of $<$? (re-

mark that if "zero" and "successor" were operations of the signature of $\mathcal{T}'$, then the question would have a trivial negative answer; in this case, however, $\mathcal{P}$ no longer would be a prime model of $\mathcal{T}'$). If such a structure $\mathcal{P}'$ would exist, then hardly one might guess that $\mathcal{P}'$ is "more natural" than $\mathcal{P}$ and that one can "directly think about $\mathcal{P}'$ " without passing through $\mathcal{P}$.

We will not answer the question involved in the latter example, which seems to be hard by itself. We only aim that the example gives an idea of what is involved in the "recursiveness problem" as compared with the "abstractness problem". Nevertheless, we are going to give a definition of "recursiveness" which would make recursive the above model $\mathcal{P}$ of $\mathcal{T}'$ if some "directly recursive" model $\mathcal{P}'$ would exist. Even if such a definition may be considered "problematic", we believe that it allows to capture basic aspects to be taken into account: in particular, despite the first example and the unanswered question involved in the second one, we will show that our formalization does not collapse, i.e., there are axiomatizable theories without recursive models.

Now we come to our formalization.

DEF. 5.1 - A model $\mathcal{M}$ of a theory $\mathcal{T}$ is called directly recursive iff the two following conditions are satisfied:

- a) for every carrier M_s (s the sort) of $\mathcal{M}$, either M_s is finite, or M_s coincides with the set N of the natural numbers;
- b) all operations and relations defined on $\mathcal{M}$ are general recursive.

DEF. 5.2 - A model $\mathcal{M}$ of a theory $\mathcal{T}$ is said to be recursive iff there are a directly recursive model $\mathcal{M}'$ of $\mathcal{T}$ and an isomorphism $i: \mathcal{M} \rightarrow \mathcal{M}'$.

= Our definition is quite in line with the two above examples; in particular, the isomorphism i does not need to be recursive.

= The choice of N as a standard carrier set corresponds to the usual way of treating the question in the general Theory of Recursiveness. From the point of view of Computer Science and, more specifically, of Data Implementation, one can find good reasons to criticize our choice. For, on one hand such a kind of representation may be quite unnatural for particular structures, where the theory itself suggests more appropriate carriers (to put the question in another way, the aim of a "theory of data" is to capture the specific features of every kind of data without passing through the natural numbers, which are particular data); on the other hand, if complexity aspects are taken into account, special representations depending on the kind

of structure may heavily lower the complexity of the operations and of the relations. However, we are only at a starting point: at the present state of research, the problem is still to capture the recursiveness of abstract data types with a level of generality comparable with the one of the Theory of Recursiveness.

= An equivalent, but slightly "more concrete" choice is made by substituting N with the set $\{0,1\}^+$ of all strings on the alphabet $\{0,1\}$. From a practical point of view, the latter choice may be convenient in special cases, but is nearly affected by the same general difficulties as the previous one.

Now we show that our definition of recursiveness does not collapse.

Th. 5.1 - There are consistent and axiomatizable theories without recursive models.

Proof. We will define a one sorted theory $\mathcal{T}$ whose signature contains exactly the nullary operation-symbol (constant) 0 (zero), the unary operation-symbol s (successor) and the unary relation-symbol P . In order to give the axioms of $\mathcal{T}$, we recall the following well known result (see e.g. [16]): two disjoint recursively enumerable sets (of natural numbers) A and B are said to be recursively inseparable iff there is no recursive set C such that $A \subseteq C$ and $B \subseteq N-C$; there are infinitely many couples of recursively inseparable sets.

Now, let $\bar{A}$ and $\bar{B}$ be two recursively inseparable sets; the axioms of $\mathcal{T}$ are the following: 1) $\forall x (\neg s(x)=0)$; 2) $\forall x \forall y (s(x)=s(y) \rightarrow x=y)$; 3) $P(s^k(0))$ for every $k \in \bar{A}$ (axiom schema); 4) $\neg P(s^h(0))$ for every $h \in \bar{B}$ (axiom schema).

Clearly, $\mathcal{T}$ is consistent and, since $\bar{A}$ and $\bar{B}$ are recursively enumerable, $\mathcal{T}$ is axiomatizable. On the other hand, all models of $\mathcal{T}$ are infinite and can be divided in two classes: a) uncountable models, which are trivially non recursive; b) countable models.

Let $\mathcal{M}$ be any countable model of $\mathcal{T}$: $\mathcal{M}$ contains a "standard submodel" $\widetilde{\mathcal{M}}$ whose carrier is the set of the natural numbers. Let $\mathcal{M}$ be recursive: then there is a directly recursive model $\mathcal{M}'$ isomorphic to $\mathcal{M}$; $\mathcal{M}'$ contains the submodel $\widetilde{\mathcal{M}'}$ which, in the considered isomorphism, is isomorphic to $\widetilde{\mathcal{M}}$; let (the infinite set of natural numbers) $\widetilde{M'}$ be the carrier of $\widetilde{\mathcal{M}'}$. Clearly, $\widetilde{M'}$ contains an element $\overline{m}_0$ which represents the constant 0, so that, being the interpretation of s in $\mathcal{M}'$ recursive, for every natural number n we can recursively find the element $\overline{m}_n$ of $\widetilde{M'}$ representing n ; also, we can recursively decide whether or not $P(\overline{m}_n)$ holds in $\widetilde{\mathcal{M}'}$. Let $\overline{C} = \{n / \widetilde{\mathcal{M}'} \models P(\overline{m}_n)\}$: clearly, $\overline{C}$ is a recursive set separating $\bar{A}$ and $\bar{B}$, a contradiction.

A kind of "non collapse" proof for our definition has been given in [2], by

showing that there is a finitely axiomatizable equational algebra whose initial model is not recursive: the proof uses the unsolvability of the word problem and refers to an equational algebra whose models are semigroups; of course, even if the initial model is not recursive, the considered algebra admits recursive models. Now, an inspection to the proof of Th.5.1 shows that the model whose carrier coincides with the set of the natural numbers and where the predicate P holds exactly on the set A is simultaneously an initial and monoinitial model of $\mathcal{T}$. By adding some axioms to $\mathcal{T}$, it is also possible to construct a theory $\mathcal{T}'$ without recursive models, with a model which is initial and monoinitial at the same time and with a non isomorphic model which is final and epifinal at the same time; more generally, the technique of the recursively inseparable sets allows us to state the following theorem, which considerably strengthens Th.5.1.

Th.5.2 - (a) One can define an axiomatizable theory without recursive models, with a model which is both initial and monoinitial and with a non isomorphic model which is both final and epifinal.

(b) One can define an axiomatizable theory without recursive models, with infinitely many non isomorphic terminal models and with infinitely many non isomorphic countable weakly monoinitial models.

Now we come to the second question, i.e. to the problem of providing a general notion of "recursive abstract data type". We start by recalling the notion of "isomorphic embedding" [4,7]; to be short, by a counter-relation of a model we will mean any complementary of any relation of that model.

DEF.5.3 - An isomorphic embedding $\varphi: \mathcal{M} \longrightarrow \mathcal{M}'$ of the model $\mathcal{M}$ of the theory $\mathcal{T}$ into the model $\mathcal{M}'$ of $\mathcal{T}$ is any homomorphism from $\mathcal{M}$ to $\mathcal{M}'$ which preserves (also) the counter-relations of $\mathcal{M}$.

Of course, every isomorphic embedding is an injective homomorphism, a fortiori it is a monomorphism; on the other hand, in presence of theories with relation symbols different from identity, an injective monomorphism is not necessarily an isomorphic embedding.

As far as we know, in a purely categorial approach there is no simple and general way to distinguish the isomorphic embeddings from the other homomorphisms; the only thing one can do is to exclude the other homomorphisms, i.e., to consider categories of models where all arrows are interpreted as isomorphic embeddings.

DEF.5.4 - A model $\mathcal{M}$ of a theory $\mathcal{T}$ is said to be isoinitial (respectively, prime)

iff for every model $\mathcal{M}'$ of $\mathcal{T}$ there is a unique (respectively, at least one) isomorphic embedding $\varphi: \mathcal{M} \rightarrow \mathcal{M}'$.

=The term "isoinitial" has been introduced in [2]; in [1] this notion is denoted by the misleading term "monoinitial", which in the present paper has a quite different meaning. Since an isomorphic embedding is a monomorphism, the only thing one can say is that an isoinitial (as well as a prime) model is weakly monoinitial; in this line, one can show that there are theories with infinitely many non isomorphic weakly monoinitial models, one of which is isoinitial.

=The weaker notion of "primeness" is the one usually considered in the so called "eastern school of Model Theory" (i.e. in Robinson's school, as it is explained in [7]). In the "western school" (to which [4] belongs), "primeness" is a very strong property which requires the notion of "elementary embedding".

=By the downward Loewenheim-Skolem Theorem and by the injectivity of the isomorphic embeddings, prime models of theories with a countable language are necessarily countable.

=A isoinitial model is unique up to isomorphisms (abstractness is captured). On the other hand, simple primeness does not implies uniqueness: our Example 2 provides a theory with infinitely many non isomorphic prime models.

=Just as it happens for epifinal and terminal models and for monoinitial and weakly monoinitial models, if a theory $\mathcal{T}$ has a isoinitial model $\mathcal{M}$ and a prime model $\mathcal{M}'$, then $\mathcal{M}$ and $\mathcal{M}'$ are isomorphic.

In [1] isoinitial models are proposed as "the formal explanation of the intuitive notion of abstract data type". This has been done not only in view of the fact that, as we are going to see, isoinitial models of axiomatizable theories turn out to be recursive, but also on the basis of a "semantical characterization theorem" (Theorem 5.1) which is stated after an intuitive analysis of the notion of "datum". The theorem requires a suitable construction, we want^{to} briefly recall in an extended form covering also theories with relation-symbols different from identity (which in [1] were not taken into account).

= Given a theory $\mathcal{T}$, a concrete datum is any quantifier-free formula $\Delta(x, y)$ (y a set of variables, x a single variable) such that $\mathcal{T} \vdash \exists ! x \exists y \Delta(x, y)$ (" $\vdash$ " means the provability relation).

= For any relation-symbol R of $\mathcal{T}$ (including the identity relation), if n is the arity of R , the n -tuple of concrete data $\langle \Delta_1, \dots, \Delta_n \rangle$ is said to be R-congru-

ent iff either $\mathcal{T} \vdash \exists x_1 \dots \exists x_n \exists y_1 \dots \exists y_n (\Delta_1(x_1, y_1) \wedge \dots \wedge \Delta_n(x_n, y_n) \wedge R(x_1, \dots, x_n))$
 or $\mathcal{T} \vdash \exists x_1 \dots \exists x_n \exists y_1 \dots \exists y_n (\Delta_1(x_1, y_1) \wedge \dots \wedge \Delta_n(x_n, y_n) \wedge \neg R(x_1, \dots, x_n))$.

= We say that $\Delta_1 \approx \Delta_2$ iff $\mathcal{T} \vdash \exists x \exists y_1 \exists y_2 (\Delta_1(x, y_1) \wedge \Delta_2(x, y_2))$. Of course, $\approx$ is an equivalence relation on the set of all concrete data.

= $\mathcal{T}$ is said to almost admit an abstract data type iff, for every relation symbol R , say n its arity, every n -tuple $\langle \Delta_1, \dots, \Delta_n \rangle$, is R -congruent.

= Let A be the set of all the equivalence classes of concrete data with respect to $\approx$, i.e. $A = \{\dots \Delta \dots\} / \approx$ (A is a set of equivalence classes in a many sorted theory, but here, for sake of simplicity and without a substantial loss of generality, we assume that $\mathcal{T}$ is one sorted). If $\mathcal{T}$ almost admits an abstract data type and A is not empty, we can define the structure $\mathcal{A}_{\mathcal{T}}$ in the following way:

i) the carrier of $\mathcal{A}_{\mathcal{T}}$ is A ;

ii) for every functional symbol f of $\mathcal{T}$, say n its arity, we set

$$f([\Delta_1], \dots, [\Delta_n]) = [\Delta(z, x_1, \dots, x_n, y_1, \dots, y_n)], \text{ where}$$

$\Delta =_{\text{df}} z=f(x_1, \dots, x_n) \wedge \Delta_1(x_1, y_1) \wedge \dots \wedge \Delta_n(x_n, y_n)$ is a concrete datum, because $\mathcal{T} \vdash \exists! z \Delta(z, x_1, \dots, x_n, y_1, \dots, y_n)$;

iii) for every relation symbol R , say n its arity, $R([\Delta_1], \dots, [\Delta_n])$ is true iff

$$\mathcal{T} \vdash \exists x_1 \dots \exists x_n \exists y_1 \dots \exists y_n (\Delta_1(x_1, y_1) \wedge \dots \wedge \Delta_n(x_n, y_n) \wedge R(x_1, \dots, x_n)),$$

which is well posed by the R -congruence hypothesis.

= A theory $\mathcal{T}$ almost admitting an abstract data type is said to admit an abstract data type iff $\mathcal{A}_{\mathcal{T}}$ is a model of $\mathcal{T}$; in this case, $\mathcal{A}_{\mathcal{T}}$ is said to be the abstract data type on $\mathcal{T}$.

Now, Th.5.1 of [1] can be extended to the present context, i.e.

Th.5.3 - A theory $\mathcal{T}$ admits an abstract data type iff there is an isoinitial model of $\mathcal{T}$; if so, the structure $\mathcal{A}_{\mathcal{T}}$ is such a model.

In view of the above characterization, every element of an isoinitial model $\mathcal{M}$ of a theory $\mathcal{T}$ can be "unambiguously" represented by a quantifier-free formula provable in $\mathcal{T}$: this amounts to say that a "unique concrete representation" of such an element can be found in every model $\mathcal{M}'$ of $\mathcal{T}$, whether or not $\mathcal{M}'$ is countable. In other words, the elements of an isoinitial model can be seen as model theoretic invariants in the sense of Kreisel [8]; this corresponds, we believe, to the maximum of abstractness.

The "unambiguity" is due to the uniqueness of the isomorphic embedding between

the isoinitial model and any model of $\mathcal{C}$ and gives rise to the following theorem:

Th. 5.4 - The (unique up to isomorphisms) isoinitial model of any axiomatizable theory (if it exists) is recursive).

Going back to our previous discussion about the adequacy of N as a standard carrier set, by Th.5.3 it is certainly more impressive to "concretely represent" every element of an isoinitial model by a quantifier-free formula (concrete datum, in the sense explained above) chosen in a suitable way in the equivalence class corresponding to such an element. Unfortunately, the axiomatizability of a theory in general implies only the recursive enumerability of the set of all concrete data and of every equivalence class, not their recursiveness.

Finally, we remark that, as far as we know, every structure that can be reasonably considered an abstract data type can be formalized as an isoinitial model: Peano Arithmetic, stacks, queues etc.

According to the above, in the choice of the isoinitial models as abstract data types there is no incompatibility between "abstractness" and "recursiveness", since it is just the "maximum of abstractness and unambiguity" which captures recursiveness. To better illustrate this point, we give a theorem showing that the presence of ambiguity in the isomorphic embeddings may give rise to non recursive models.

Th.5.5 - There are axiomatizable theories without recursive models and with infinitely many non isomorphic and non recursive prime models.

For a given non recursive prime $\mathcal{P}$, one can cut off ambiguity by adding to $\mathcal{T}$ the diagram $D_{\mathcal{C}}(\mathcal{P})$: in this case $\mathcal{P}$ becomes isoinitial in the set of models of the theory $\mathcal{T}' = \mathcal{T} \cup D_{\mathcal{C}}(\mathcal{P})$, while any other (possible) prime model $\mathcal{P}'$ of $\mathcal{T}$ which is not isomorphic to $\mathcal{P}$ is no longer a model of $\mathcal{T}'$; but even if the set of constants C is recursive, the diagram $D_{\mathcal{C}}(\mathcal{P})$ cannot be recursive; thus, the theory is no longer axiomatizable.

REFERENCES

- [1] Bertoni, A., Mauri, G., Miglioli, P., A characterization of abstract data as model theoretic invariants, Proc. 6° ICALP, Lect. Not. in Comp. Sci., 71, pp. 26-37, Springer, Berlin, 1979
- [2] Bertoni, A., Mauri, G., Miglioli, P., Wirsing, M., On different approaches to abstract data types and the existence of recursive models, Bulletin EATCS, 9, 1979

- [3] Broy, M., Dosch, W., Partsch, H., Pepper, P., Wirsing, M., Existential quantifiers in abstract data types, Proc. 6^o IICALP, Lect. Not. in Comp. Sci. 71, 73-87, Springer, Berlin, 1979
- [4] Chang, C.C., Keisler, H.J., Model theory, North-Holland, Amsterdam, 1973
- [5] Goguen, J.A., Thatcher, J.W., Wagner, E.G., An initial algebra approach to the specification, correctness, and implementation of abstract data types, IBM Res. Rep. RC6487, Yorktown Heights, 1976
- [6] Guttag, J.V., Horning, J.J., The algebraic specification of abstract data types, Acta Informatica 10, 27-52, 1978
- [7] Keisler, H.J., Fundamentals of model theory, in "Handbook of Mathematical Logic" (J. Barwise ed.), 47-104, North-Holland, Amsterdam, 1977
- [8] Kreisel, G., Model-theoretic invariants: applications to recursive and hyperarithmetic operations, Proc. Symp. on the Theory of Models, North-Holland, Amsterdam, 1965
- [9] Liskov, B.H., Zilles, S.N.; Programming with abstract data types, SIGPLAN Notices 6, 1974
- [10] Mac Lane, S., Categories for working mathematician, Springer, Berlin, 1971
- [11] Majster, M.E., Data types, abstract data types and their specification problem, Theoretical Computer Science 8, 89-127, 1979
- [12] Morris, F.L., Types are not sets, Proc. ACM Symp. On the Principles of Programming Languages, 120-124, 1973
- [13] Rogers, H., Theory of recursive functions and effective computability, Mc Graw Hill Book Co, New York, 1967
- [14] Standish, T.A., Data structures: an axiomatic approach, in "Current trends in programming methodology" (R.T. Yeh ed.), vol.IV, Prentice Hall, 1978
- [15] Zilles, S.N., Algebraic specification of data types, Project MAC Progress Rep. 11, MIT, Cambridge, Mass., 1974
- [16] Wand, M., Final algebra semantics and data type extensions, Indiana University, Comp. Sci. Dept., Tec. Rep. n.65; 1977

Acknowledgments: we wish to thank M. Wirsing, for having stimulated a comparative analysis of various formalizations of "abstract data types", and for helpful discussions about the notion of "recursiveness" of abstract data types.