Semi-constructive formal systems and axiomatization of abstract data types

Pierangelo Miglioli, Ugo Moscato, Mario Ornaghi
Department of Information Science - University of Milan

1. Introduction

In the area of abstract data types (ADT) the "isoinitial model" approach [2,3,4] based on (classical) model theory, has been proposed with two aims: to provide a simple treatment of the recursiveness problem and to allow ADT specifications less restrictive than the "algebraic" ones [6,10,12].

As for the latter aspect, the algebraic attitude is mainly interested in setting up "small" theories (i.e., with a small deductive power as compared, e.g., with the one of full first order arithmetic) in order to axiomatize simple ADT to be furtherly extended in a (possibly long) sequence of "small" refinement steps. This point of view, which has given rise to important developments, is surely adequate to (stepwise) program synthesis; however, it is not oriented to the so called "constructive" attitude [1,5,8,9,11,16,17,18,19,22] which looks at "proofs" (of a constructive formal system) as "programs".

On the contrary, the isoinitial approach is quite in line with the latter attitude. Accordingly, in [4] the problem has been considered of characterizing constructive formal systems providing, together with a powerful ADT specification-method, a set of "abstract" algorithms on the ADT, i.e., the set of (constructive) proofs definable in them. Of course, this requires "reasonably powerful" formal systems (i.e., formal systems with a deductive power comparable, e.g., with the one of intuitionistic first order arithmetic).

In this frame, the possibility of looking for "large" *constructive* and *classically sound* formal systems S=T+L has been considered by the authors, where:

- the notation "T+L" means that the system S consists of a mathematical part T (a first order theory in the sense of [7] which, interpreted according to classical semantics, has an isoinitial model [3,4]) and a superintuitionistic logic L (i.e., INT$\subseteq$L$\subseteq$CL, INT and CL being intuitionistic and classical logic respectively);
- the sense according to which S is *constructive* is that S satisfies the disjunction property DP and the explicit definability property EDP [20,21];

- the sense according to which *classical soundness* is assumed is that T+L is consistent iff T+CL is.

As it is known, there are constructive first order systems which are consistent but not classically consistent [20]. Perhaps, as far as only operational (procedural) interpretations are considered (where a logical formula is read as something as a λ-expression, see, e.g., [8,13,16]) the classical soundness is not important. But we are interested, as said before, in connecting the area of constructivism with the one of ADT: and the latter has been developed (by means of algebraic or model theoretic tools) in a classical context.

Also, we believe that the classical reading of formulas has a simplicity which hardly can be found in other kinds of interpretations; this simplicity (which has been taken as paradigmatic in fields of computer science such as program correctness and artificial intelligence) makes classical semantics the most natural "denotational semantics" (to be preserved by an operationally correct formal system).

According to the above, our "large" systems correspond to the attempt of setting up as great as possible recursively axiomatizable constructive systems S=T+L contained in the classical T-system T+CL, even if the following limitations cannot be avoided:
- the greatest constructive subsystem (for a given T) of T+CL doesn't exist [14,21] (there is a set of maximal, "constructively incompatible" systems);
- the maximal constructive subsystems of T+CL are not, in general, recursively axiomatizable (even if T+CL is).

In this line, some results oriented to a classification (concerning the mutual "constructive compatibility" of various powerful logical and mathematical principles) have been obtained by the authors. Also, classically consistent systems corresponding to a weaker notion (we have called "semi-constructiveness") have been found, where S=T+L is *semi-constructive* iff it satisfies the following *weak disjunction property* WDP and *weak explicit definability property* WEDP:

(WDP) $S \vdash A \vee B$ and $A \vee B$ is closed $\Rightarrow$ $T+CL \vdash A$ or $T+CL \vdash B$
(WEDP) $S \vdash \exists x C(x)$ and $\exists x C(x)$ is closed $\Rightarrow$ $T+CL \vdash C(t)$ for some closed t.

We call *sub-constructive* any system S=T+L contained in some (fully) constructive system S'=T+L' (with the **same** mathematical part T): it turns out that any sub-constructive system is semi-constructive (but the converse does not hold, as we will see).

We look at sub-constructiveness and, more generally, at semi-constructiveness as one of the two extremes within which a constructive point of view may range, the other being strong constructiveness, where (roughly speaking): by a *strongly constructive system* we will mean any system S such that any proof of $A \vee B$ in S ($A \vee B$ closed) contains sufficient information to build up a proof of A in S or a proof of B in S, and the like for $\exists x A(x)$.

Strong constructiveness is appropriately treated in a proof theoretical attitude, while sub-constructiveness and semi-constructiveness generally involve simpler model theoretic aspects. Form the point of view of program specification and construction the former is needed in contexts such as program synthesis, where proofs are taken as programs and must give rise to effective computations; on the other hand, the latter can be used if one is not interested in computational devices, but only in foreseeing (as a first approximation) that some functions or relations (definable in the frame of first order theories) are (in principle) computable, or in guaranteeing that some expansions of given "intended models" satisfy some general requirements (having a character more semantical than syntactical). In this sense, a notion such as constructiveness (as defined by DP and EDP), turns out to be, according to the cases, undercharacterized or overcharacterized, while sub-constructiveness and semi-constructiveness seem to be adequate to ADT specification and extension.

Thus, one of the aims of the present paper is to show how semi-constructiveness can be used in a stepwise method of ADT-definition. In particular, we will provide a rather general result (see THEOR.4 below) allowing to pass from a theory T with an isoinitial model M to a stronger T' with an isoinitial model M' expanding M. Results of this kind have been expounded in [4] only for definitory extensions of T; here we will provide a criterion allowing to obtain from T a T' which may be *non conservative* over T.

Also, we will discuss how these results can be used in the definition of classes of ADT, i.e. of families of ADT together with instantiation and extension methods (as typical in an "object oriented" attitude).

The more powerful the semi-constructive systems one uses are, the more powerful our extension method becomes. Thus, the development of axiomatic methods to set up great semi-constructive systems not only is interesting from a purely logical point of view but also might give rise to applications in the area of ADT. In this line, in the last part of the paper we will present five classes of (mutually "incompatible") or sub-constructive systems coming from our classification; this is only a first classification and we hope to find examples where the listed principles are useful.

2. The logic IKA and theories completely formalizing an ADT

In this section we briefly introduce two basic notions involved in the paper; a more detailed treatment can be found in [4] (where IKA is called CON) and in [3].

As said above, we are interested in semi-constructive and classically sound systems S=T+L. The classical soundness is automatically guaranteed by using superintuitionistic logics L containing the logic IKA so characterized: IKA is obtained by adding to INT (intuitionistic predicative logic with identity) the following axioms:

(K) $\forall x \sim\sim A(x) \rightarrow \sim\sim \forall x A(x)$ (Kuroda principle)

(A) $\sim\sim A \rightarrow A$ for any *atomic* A

Of course, the restriction to the atomic formulas in axiom (A) prevents the validity (in IKA) of the non constructive principle
$\sim\sim H \rightarrow H$ for any H.

We remark that the addition of (K) alone to INT is sufficient to obtain classically consistent systems.

We say that a model M of a theory T is *isoinitial in T* [2,3,4] iff, for every model M' of T, there is a unique isomorphic embedding [7] from M to M'. For instance, the standard structure of natural numbers is an isoinitial model of Peano Arithmetic and is the "intended" model of Peano axioms. We give the following definition:

DEF.1 A theory T completely formalizes an ADT iff T has a reachable isoinitial model M.

If an isoinitial model I of T exists, then: (1) I is recursive [2,3,4]; (2) any other isoinitial model of T is isomorphic to I; (3) T can be extended with the addition of a recursive diagram into a theory T' completely formalizing an expansion (with new constants) of I. By (2) we can choose any isoinitial M in order to represent the ADT formalized by T; we will say also that T formalizes the ADT I. We say that T is *atomically complete* iff T+CL $\vdash A$ or T+CL $\vdash \sim A$, for every closed atomic formula A.

The following theorem, whose proof is implicitly contained in [3,4], gives an useful criterion to study isoinitiality.

THEOR.1. T completely formalizes an ADT iff T has a reachable model and T is atomically complete.

In the above, we don't impose any restriction on the form of the axioms and we consider also many sorted first order languages. THEOR.1 allows to prove:

PROP.1. For every set C of "constructors" (i.e. constant and function symbols), the term-algebra generated by C is an isoinitial model of the theory T(C) = identity + injectivity axioms + induction principles.

The injectivity axioms state that different closed terms represent different objects. Identity and injectivity axioms are sufficient to obtain the isoinitiality result, but it is useful to introduce also various induction principles in order to use logical semi-constructive systems in a relevant way; in particular, we consider the usual structural induction and the descending chain principle based on the well founded order relation related to the structural complexity of the closed terms [4,16].

3. Towards an ADT-definition methodology within semi-constructive systems

Our approach to ADT is based on the previous notion of a theory completely formalizing an ADT and on THEOR.1. In this frame, we propose a method where the first order axioms characterizing an ADT are built up in one or more steps, in such a way that each step satisfies the requirements of THEOR.1. In particular, it may happen that condition of atomic completeness of THEOR.1 can be proved by showing that a suitable formula of the kind $\forall x(H(x) \vee \sim H(x))$ can be proved in a semi-constructive system T'+L (for a suitable $T' \supset T$ and a suitable logic L). Also, a stepwise construction can be given for *classes* of ADT, as we will see later.

In the first step of the construction of an ADT one essentially uses THEOR.1. In particular, PROP.1 proposes a general way to start, but, of course, other theories T satisfying THEOR.1 can be taken. The subsequent steps are *extension steps,* where the extension of an ADT by new relations, functions and sorts is characterized by the following definition:

DEF.2 Let T be a theory completely formalizing an ADT M, let LT be the language of T and let LT' be a language extending LT by new sort, relation and function symbols; we say that a theory $T' \supseteq T$ is an ADT-extension of T into LT' iff T' completely formalizes an ADT M' which is an *expansion* [7] of M into LT' (i.e. M' interprets the symbols of LT *exactly* as M and the old carriers are unchanged).

The requirements of DEF.2 are analogous to the safety conditions required to preserve "sufficient completeness" in the algebraic attitude [12].

Using THEOR.1, we can easily prove the following result.

THEOR.2. Let T be a theory in a language LT; let S be a sort symbol not in LT and $c_0,..,c_n,f_0,..,f_k$ be constant and function symbols of sort S not in

LT (the arities of the new function-symbols may contain also sorts of LT); let LT' be the extension of LT by $S,c_0,.,c_n,f_0,.,f_k,=$ (= the identity on S) and T'=(T + identity and injectivity ax. + induction principles for the new symbols); then T' is an ADT-extension of T into LT'.

This theorem allows the enrichment by new sorts and also the definability of a class of parametric ADT (indeed, the starting theory T may be any theory completely formalizing an ADT).
To add new functions and relations, one can use explicit definitions [7]; but it may happen that an extension by explicit definition is *not* an ADT-extension. We give the following theorem (stated in [17] for particular systems based on the logic IKA and here extended to *any* semi-constructive system):

THEOR.3 Let T be a theory completely formalizing an ADT, let LT be the language of T and let L be an intermediate logic. If the system S=T+L is semi-constructive, then: (I) let us consider the definition axiom $\forall x(r(x) \leftrightarrow H(x))$ (r a new relation symbol, H any formula); if $S \vdash \forall x(H(x) \vee \sim H(x))$, then the theory $T'=T\cup\{\forall x(r(x)\leftrightarrow H(x))\}$ is an ADT-extension of T into $LT\cup\{r\}$; (II) let us consider the definition axiom $\forall xF(x,f(x))$ (f a new function symbol, F any formula); if $S\vdash\forall x\exists!zF(x,z)$, then the theory $T'=T\cup\{\forall xF(x,f(x))\}$ is an ADT-extension of T into $LT\cup\{f\}$.

If the starting theory T is too poor, generally one cannot characterize interesting functions by explicit definitions. To obtain a more general result, involving also *non conservative* extensions, the following definitions are in order:

A is *existentially compound* (**e.c.**) iff one of the following clauses applies:

1) A=B and B is quantifier free;
2) $A=B\wedge C$ or $A=B\vee C$, and B, C are **e.c.**;
3) $A=\exists xB$ and B is e.c.;
4) $A=B\rightarrow C$, C is e.c. and $B=\forall \mathbf{x}D$, where D is quantifier free and "$\forall \mathbf{x}$" indicates a possibly empty sequence of universal quantifications.
A *good* formula is any formula of the kind $\forall \mathbf{x}(A(\mathbf{x})\rightarrow B(\mathbf{x}))$ where $A(\mathbf{x})$ is e.c. and $B(\mathbf{x})$ is quantifier free.

Now we can prove:

THEOR.4 Let T, LT and L be defined as in THEOR.3. We have: (I) let r be a new relation symbol, let $LT'=LT\cup\{r\}$, let $ax_r\subseteq LT'$ be a set of *good* formulas and let $T'=T\cup ax_r$; if T is consistent, S=T'+L is semi-constructive and $S \vdash \forall x(r(x) \vee \sim r(x))$, then T' is an ADT-extension of T into LT'; (II)

let F be a new relation symbol, let f be a new function symbol, let $LT'=LT\cup\{F\}$ and $LT''=LT\cup\{F,f\}$; let $T'=T\cup ax_F$, where ax_F is a set of *good* formulas of LT'; if T' is consistent, $S=T'+L$ is semi-constructive and $S \vdash \forall x\exists!zF(x,z)$, then $T''=T'\cup\{\forall xF(x,f(x))\}$ is an ADT-extension of T into LT''.

We remark that the use of powerful semi-constructive systems according to THEOR.4 allows to capture, *within the formal systems themselves,* aspects which in the above quoted algebraic attitude can be expressed only at a metatheoretical level.

We also remark that Theor.'s 3 and 4 *still hold* if one takes as ADT the *initial* models [3,10] instead of the isoinitial ones (on the other hand, as it happens for isoinitiality, arbitrary explicit definitions don't preserve initiality).

An example of ADT defined by extensions is:

(E1) Peano Arithmetic can be obtained starting with the set of constructors $C=\{0,s\}$ [so the theory T(C) contains the injectivity axioms $\forall x(\sim s(x)=0)$, $\forall x\forall y(s(x)=s(y)\rightarrow x=y)$ and the usual induction schema] and *extending* it by + and * using THEOR.4 and the logic IKA.

An extensive analysis of the applicability of semi-constructive systems stronger that IKA in ADT-extensions requires a great amount of "experimental" work; we are at the beginning in this direction.

However, we have sketched some schemas which seem to be promising. The following example shows a general way of applying the principle:

(WGRZ) $\sim\sim\forall xB(x)\wedge\forall x(A\vee B(x)) \rightarrow A\vee\forall xB(x)$

which is contained in all the sub-constructive systems we will present in the next section.

(E2) Let us assume that $\forall x\exists!yB(x,y)$ can be classically proved from a T completely formalizing an ADT (as said above, the introduction in T of a corresponding explicit definition introducing a new function-symbol may not preserve isoinitiality or initiality).

There are cases where one can prove (e.g. by suitable inductive principles) that $T+L \vdash \forall x(A\vee\exists yB(x,y))$, for suitable formulas A and suitable logics L which contain (WGRZ) and are such that T+L is sub-constructive. In these cases, an application of (WGRZ) provides $T+L\vdash A\vee\forall x\exists yB(x,y)$.

Now, let $\sim B$ be consistent with T: then, by the subconstructiveness of T+L, one deduces that there is a logic L' (possibly non effective) such that $L\subseteq L'$, T+L' is constructive and $T+L' \vdash \forall x\exists!yB(x,y)$.

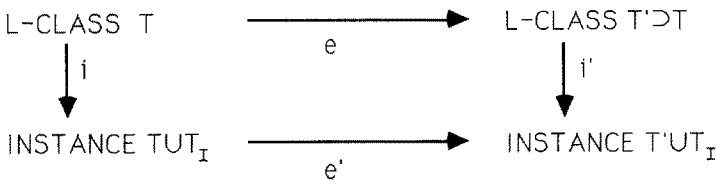
Even if in general one cannot use T+L' to compute the defined function, one can be sure that the addition of the considered explicit definition to T preserves isoinitiality (preserves initiality).

Now, we come to the notion of a *class of ADT*. Firts of all, we remark that, if we have a theory T satisfying the conditions of THEOR.1 and we add to T the axioms (*) $\forall x(r(x) \vee \sim r(x))$ for any relation symbol r, we obtain a $T' \supset T$ which, of course, completely formalizes an ADT (indeed, (*) is classically valid; thus, neither the class of the models of T nor the set of the classical theorems of T are affected). But formulas such as (*) become relevant axioms whenever one considers semi-constructive systems T+L instead of T+CL, as we are going to explain.

A theory T *formalizes a L-class* (or, simply, T *is a L-class*) iff the formulas (*) can be proved in T+L. A L-class T is *not* required to formalize an ADT and T+L *doesn't need* to be sub- or semi-constructive.

An *instance* of a L-class T is any $I = T_I \cup T$ such that I completely formalizes an ADT; we call T_I an "instantiation" for T. We don't require any constructiveness property on I; according to the cases, it may be semi-constructive, sub-contructive, strongly constructive or nonconstructive at all.

Let T be a L-class: we say that $T' \supset T$ is a *L-class-extension* of T iff T' is an L-class (expanding T by new sorts, functions or relations) and, for every instantiation T_I for T and T', $T' \cup T_I$ is an ADT-extension of $T \cup T_I$. I.e., one has the following commutative diagram:



where e is a class-extension, i and i' are provided by the **same** instantiation T_I and e' is the ADT-extension corresponding to e, T_I.

Now, one can see that the formulas (*) are useful to prove that a $T' \supset T$ is a *class-extension* of T; such a proof can be based on the above theorems and on the provability of the suitable formulas in T'+L.

The following example briefly shows an IKA-class and an IKA-class extension.

(E3). IKA-CLASS T_{seq} (finite sequences of generic elements).

CONSTRUCTORS: nil : seq; a : seq, elem -> seq;

RELATIONS: = : seq, seq; = : elem, elem; ≤ : elem, elem;

IDENTITY AXIOMS: the usual ones;

INJECTIVITY AXIOMS: (i1) $\forall x \forall y \sim nil = a(x,y)$;

(i2) $\forall x_1 \forall x_2 \forall y_1 \forall y_2 (a(x_1,y_1)=a(x_2,y_2) \rightarrow x_1=x_2 \wedge y_1=y_2)$.

IKA-CLASS AXIOMS: (c1) $\forall x \forall y(x = y \vee \sim x = y)$; (c2) $\forall x \forall y(x \leq y \vee \sim x \leq y)$.

STRUCTURAL INDUCTION RULE: $\dfrac{H(nil) \qquad H(a(j,x))}{H(t)}$

The formula $\forall x_1 \forall x_2(x_1 = x_2 \vee \sim x_1 = x_2)$ can be proved by induction starting from (c1) and the other axioms.

For every T_{elem} with (at least) a sort elem, if T_{elem} completely formalizes an ADT (and if the obvious requirement to merge the languages of T_{elem} and T_{seq} in a sound way is satisfied) then $T_{elem} \cup T_{seq}$ completely formalizes an ADT (i.e., $T_{elem} \cup T_{seq}$ is an instance of T_{seq}).

An IKA-class-extension of T_{seq} can be obtained , e.g., by adding suitable axioms (ORD) defining an ordering $\leq$ on sequences induced by the ordering $\leq$ on elements; to be sure that $T'_{seq} = T \cup (ORD)$ is powerful enough to provide an IKA-class-extension, one has to prove in T'_{seq}+IKA the formula $\forall x_1 \forall x_2(x_1 \leq x_2 \vee \sim x_1 \leq x_2)$.

4. Some semi-constructiveness results

Here we explain some results which relate the form of the axioms of T to the logic L in order that S=T+L be semi-constructive. To do so, first of all we introduce the following superintuitionistic logical principles:

(MA) $\forall x(A(x) \vee \sim A(x)) \wedge \sim\sim \exists x A(x) \rightarrow \exists x A(x)$;

(KP$\vee$) $(A \rightarrow B \vee C) \rightarrow (A \rightarrow B) \vee (A \rightarrow C)$;

(KP$\exists$) $(A \rightarrow \exists x B(x)) \rightarrow \exists x(A \rightarrow B(x))$, with x not free in A;

(GRZ) $\forall x(A \vee B(x)) \rightarrow A \vee \forall x B(x)$, with x not free in A.

(WGRZ) $\sim\sim \forall x B(x) \wedge \forall x(A \vee B(x)) \rightarrow A \vee \forall x B(x)$, with x not free in A.

These principles, except (WGRZ), are well known: (MA) is Markov Principle [20,21]; (KP$\vee$) has been introduced by Kreisel and Putnam in order to provide a propositional logic stronger than the intuitionistic one and satisfying the disjunction property [15]; (KP$\exists$) is a predicative variant of (KP$\vee$) which is also known as (IP) [20,21]; (GRZ) is Grzegorczyk principle (see Smorinski's essay in [20]); finally, (WGRZ) is a weak variant of (GRZ) which turns out to have a much wider applicability than the former (differently from (GRZ), it can be combined with theories containing induction principles, as we will see).

Some of the above principles are separately meaningful for program synthesis (e.g., (MA) allows to deal with the minimalization operator), but not all are "constructively compatible": more precisely, (MA)+(KP$\exists$)+Intuit. arithmetic = Class. arithmetic [21]; also, one can show that (MA)+(KP$\vee$)+Intuit. arithmetic is not semi-constructive; finally, (GRZ) + Intuit. arithmetic=Class. arithmetic [20]. We also remark that

(GRZ), with the addition of the intuitionistic principles, allows to derive (MA).

Now, we are going to present five classes of semi-constructive systems which are mutually "constructively incompatible"; each class will contain one (or more) of the principles (MA), (KP$\vee$), (KP$\exists$) and (GRZ) together with (WGRZ) and appropriate new logical principles. We start with the following definitions:

a *$\forall$-formula* is of the kind $\forall \mathbf{x} H$, a *$\forall\exists$-formula* is of the kind $\forall \mathbf{x} \exists \mathbf{y} H$, with H quantifier free and $\forall \mathbf{x}$, $\exists \mathbf{y}$ possibly empty sequences of quantifications;

a *$\forall\exists\sim$-formula* is inductively so defined:

every $\forall\exists$-formula is a $\forall\exists\sim$-formula; every formula such as $\sim A$ is a $\forall\exists\sim$-formula; if A,B are $\forall\exists\sim$-formulas and C is any formula, then $A\wedge B$, $C\rightarrow A$, $\forall x A$ are $\forall\exists\sim$-formulas;

a *$\forall\sim$-formula* is defined likewise taking in the basic clause the $\forall$-formulas instead of the $\forall\exists$-formulas.

We remark that the set of the $\forall\exists\sim$-formulas contains both the set of the $\forall\sim$-formulas and the set of the $\forall\exists$-formulas; also, the set of the $\forall\sim$-formulas doesn't contain the set of the $\forall\exists$-formulas, and conversely; finally, the set of the $\forall\sim$-formulas contains the set of the Harrop formulas (which are defined, e.g., in [20]).

To define the first class of formal systems, we introduce the two following logical principles:

(P1) $\exists x A(x) \vee \forall x(A(x)\wedge\sim\sim B(x) \rightarrow B(x))$

(P2) $A(t) \vee \exists x(A(x)\wedge\sim\sim B(x) \rightarrow B(x))$

Now the *1-systems* T+L are so defined:

T completely formalizes an ADT;

T (possibly) contains induction principles (also in the form of descending chain principles) and all its other axioms are $\forall\exists\sim$-formulas;

L=IKA+(MA)+(WGRZ)+(P1)+(P2).

We can prove:

THEOR.5 The 1-systems are sub-constructive.

Remark 1. As far as only sub-constructiveness and, more generally, semi-constructiveness is involved, results such as THEOR.5 and the following THEOR.'s 6 and 7 can be proved in a reasonably simple way (much more complex proofs are required to establish the full constructiveness of some "large subsystems" of the ones presented here, discussed, e.g., in [17]). The proofs of our theorems can be seen simply as soundness proofs and involve the following aspects, where S=T+L is the system in hand and S_{CL}=T+CL is the corresponding classical system:

- one has to define a set F(T) of S_{CL}-provable formulas such that F(T) is closed under the IKA-inference rules and satisfies the disjunction property and the explicit definability property;
- one has to prove that all the S-provable formulas are contained in F(T).

We call F(T) a frame for the system S. The important aspect to be pointed out is that the frame is fully constructive but is not, in general, recursively enumerable; on the other hand, the recursively enumerable set of the S-provable formulas turns out to be sub-constructive but cannot be guaranteed to be (fully) constructive. As an example, we present the frame F1(T) to be used for the 1-systems considered in THEOR.5 (the frames for the systems considered in the subsequent theorems are omitted in this paper):
- if H is a closed formula, then $H \in F1(T)$ iff $T+CL \vdash H$ and one of the following clauses applies:
1) $H = \sim B$ or H is atomic;
2) $H = B \wedge C$ and $B \in F1(T)$ and $C \in F1(T)$;
3) $H = B \wedge C$ and $B \in F1(T)$ or $C \in F1(T)$;
4) $H = B \rightarrow C$ and $B \notin F1(T)$ or $C \in F1(T)$;
5) $H = \exists x B(x)$ and there is a closed term t such that $B(t) \in F1(T)$;
6) $H = \forall x B(x)$ and for every closed term t $B(t) \in F1(T)$;
- if $H = A(\mathbf{x})$ is an open formula, then $H \in F1(T)$ iff $A(\mathbf{t}) \in F1(T)$ for every $A(\mathbf{t})$ obtained from $A(\mathbf{x})$ by substituting the free variables $\mathbf{x}$ with closed terms.

One easily sees that F1(T) is closed under the inference rules and satisfies DP and EDP (one can also show that F1(T) is maximal in the following sense: if F(T) is any set of S_{CL}-provable formulas such that F(T) is closed under the IKA-inference rules, satisfies DP and EDP and contains F1(T), then F(T)=F1(T)). Then, the proof of THEOR.5 amounts to show that if S=T+L is a 1-system then all the S-provable formulas are contained in F1(T).

Remark 2. THEOR.5 can be used to extend an ADT according to THEOR.3: for, THEOR.3 requires the semi-constructiveness of the system T+L, where T is the starting theory.

On the other hand, THEOR.5 cannot be used, as such, to extend an ADT according to THEOR.4: for, one needs the semi-constructiveness of the system T'+L, where T' is the extended theory , just to be sure that T' completely formalizes an ADT; but the definition of a 1-system requires that the involved theory (here T') completely formalizes an ADT. The problem is related to the following aspects:

- the possibility of using $\forall\exists\sim$-axioms requires (in order that T'+L be semi-constructive) the atomic completeness of T' (this is an assumption weaker than the hypothesis that T' completely formalizes an ADT, but even the atomic completeness of T' is provided, in THEOR.4, by the semi-constructiveness of T'+L);
- the possibility of using (semi-constructively) (MA) (together with $\forall\exists\sim$-axioms and induction principles) requires that T' has a reachable model (i.e., since T' must be atomically complete by the above point, that T' completely formalizes an ADT).

These difficulties can be partially overcome as follows.
Let LT' be the language of a theory T' and let LT be a language contained in LT'; let M be any structure for the language LT' (not necessarily a model of T): we say that M is an LT-model of T' if all formulas of T belonging to LT hold in M; we say that T' is LT-atomically complete iff $T'\vdash A$ or $T'\vdash\sim A$ for every closed atomic formula $A\in LT$; we say that a LT-model M of T' is LT-reachable if every element of the carrier of M is denoted by some closed term of LT; finally, we say that T' completely formalizes a LT-ADT iff T' is LT-atomically complete and T' has a reachable LT-model M.
Now, the LT-LT'-1-systems T'+L' are so defined:
- $LT\subseteq LT'$, where LT' is the language of T, and T' completely formalizes a LT-ADT;
- T' (possibly) contains induction principles (in the full language LT') and all other axioms of T' are $\forall\exists\sim$-formulas of LT or Harrop-formulas (in the full language LT');
- $L'=IKA+(MA)_{LT}+(P1)+(P2)$, where $(MA)_{LT}$ is the set of all the instances of (MA) in the language LT.

Then we can prove:

(a) The LT-LT'-1-systems are sub-constructive.
If one is not interested in using $(MA)_{LT}$, then one can consider the weak LT-LT'-1-systems: they are defined as the LT-LT'-1-systems, with the only difference that $(MA)_{LT}$ is not included in L' and T' must be LT-atomically complete but may not completely formalize an ADT. One can prove:
(b) The weak LT-LT'-1-systems are sub-constructive.

To introduce other kinds of semi-constructive systems, we need the following definitions:
a formula A is *stable* iff: A is atomic or negated, or A is of the form $B\wedge C$ or $B\vee C$ or $B\rightarrow C$ or $\forall xB$ with B,C stable;

A is a *formula in* $U, U \to V$ iff A is any formula constructed starting from U, $U \to V$ and using only the propositional connectives;
a formula A is *negatively saturated* iff every (possible) quantifier occurrence in it is in the scope of $\sim$.

Also, we introduce the following families of principles:

$(Q\vee)$ $(\forall x(\sim\sim H(x)\wedge\sim\sim K(x)\to\forall y\sim U(y))\to(\forall x(H(x)\to K(x))\to C\vee D))\to$
$(\forall x(\sim\sim H(x)\wedge\sim\sim K(x)\to\forall y\sim U(y))\to$
$(\forall x(H(x)\to K(x))\to C)\vee(\forall x(H(x)\to K(x))\to D))$,
where C is any formula in U, U->V (D is arbitrary).

$(Q\exists)$ $(\forall x(\sim\sim H(x)\wedge\sim\sim K(x)\to\forall y\sim U(y))\to(\forall x(H(x)\to K(x))\to\exists w C(w)))\to$
$(\forall x(\sim\sim H(x)\wedge\sim\sim K(x)\to\forall y\sim U(y))\to\exists(\forall x(H(x)\to K(x))\to C(w)))$, where C is any formula in U, U -> V.

If in $(Q\vee)$ and $(Q\exists)$ we take a stable K(x), we obtain $(SQ\vee)$ and $(SQ\exists)$; if in $(Q\vee)$ we take a negatively saturated C, we obtain $(NSQ\vee)$.

We remark that $(SQ\vee)$ and $(SQ\exists)$ allow to deduce $(KP\vee)$ and $(KP\exists)$.

Now, the *2-systems* T+L are so defined:
T is atomically complete;
T (possibly) contains induction principles (including descending chain principles) and all its other axioms are $\forall\exists$-formulas or $\forall\sim$-formulas;

$L = IKA + (WGRZ) + (SQ\vee) + (SQ\exists)$.

The *3-systems* T+L are so defined:
T is atomically complete;
T (possibly) contains structural induction (but no descending chain principle) and all its other axioms are $\forall\exists$-formulas or $\forall\sim$-formulas;
$L = IKA + (WGRZ) + (KP\vee) + (NSQ\vee)$.

Remark 3. (NSQ) doesn't allow to deduce $(KP\vee)$.
We can prove:

THEOR.6 The 2-systems and the 3-systems are sub-constructive.

Remark 4. To apply this result to THEOR.4, we define the LT-LT'-2-systems T'+L' as follows:
for the LT-LT'-2-systems, $L'=IKA+(SQ\vee)+(SQ\exists)$, while T' must be LT-atomically complete, may contain induction principles in the full language LT' (including descending chain principles) and all its other axioms are $\forall\exists$-formulas of LT, or $\forall\sim$-formulas of LT, or Harrop-formulas (in the full language LT');
for the LT-LT'-3-systems, $L'=IKA+(KP\vee)+(NSQ\vee)$, while T' satisfies the same conditions as in the LT-LT'-2-systems, with the only difference

that descending chain principles are not allowed (structural induction, on the other hand, is allowed).

We can prove:

(c) The LT-LT'-2-systems and the LT-LT'-3-systems are sub-constructive.

The addition of (KP∃) or of (MA) to the 3-systems does not preserve semi-constructiveness; thus, the 1-systems, the 2-systems and the 3-systems are mutually "incompatible".
Logics L more powerful than the ones considered above can be (semi-constructively) used together with sufficiently weak theories T. More precisely:
- we say that a theory T is a Harrop-theory iff all axioms of T are Harrop-formulas (no general induction principles are allowed!);
- we say that a system T+L is a *4-system* if T is a Harrop-theory and L=IKA+(GRZ)+(Q∨)+(Q∃) (remark that T is not necessarily atomically complete and that (Q∨) and (Q∃) are taken without restrictions);
- we say that a system T+L is a *5-system* if T is a Harrop-theory and L=IKA+(GRZ)+(P1)+(P2) (again, T may not be atomically complete).

We can prove:

THEOR.7 The 4-systems and the 5-systems are sub-constructive.

If L is the logic IKA+(GRZ)+(Q∨)+(Q∃)+(P1)+(P2) and T is the empty theory (hence, T is a Harrop-theory), then T+L collapses, i.e., T+L is not semi-constructive. Thus, the 4-systems and the 5-systems are constructively "incompatible".

Remark 5. The systems considered in the above THEOR.'s 5, 6 and 7 can be extended into (fully) constructive (possibly non effective, i.e., non recursively enumerable) systems: examples of the latter are the corresponding frames (in the sense of Remark 1). On the other hand, one can define recursively axiomatizable semi-constructive systems S=T+L (effective or not) which are not sub-constructive. A similar situation is found if only DP, e.g., is taken into account. For instance, let L=IKA+(KP∨)+(KP∃)+A∨(A→B∨∼B) and let T be any theory satisfying the conditions of THEOR.5: then, T+L satisfies WDP; on the other hand, there is a T (in this class of theories) such that T+L cannot be extended into a system T+L' satisfying DP.

This work has been supported by grants of the Ministero della Pubblica istruzione (40% and 60%).

REFERENCES.

[1] Bates J., Constable R. - Proofs as programs - ACM Transaction on Programming Languages and Systems, vol. 7, n.1, 1985.

[2] Bertoni A., Mauri G., Miglioli P., Wirsing M. - On different approaches to abstract data types and the existence of recursive models - EATCS bulletin vol. 9, oct. 1979.

[3] Bertoni A., Mauri G., Miglioli P. - On the power of model theory to specify abstract data types and to capture their recursiveness - Fundamenta Informaticae IV.2, 1983, pp. 127-170.

[4] Bertoni A., Mauri G., Miglioli P., Ornaghi M. - Abstract data types and their extension within a constructive logic - Semantics of data types (Valbonne, 1984), Lecture Notes in Computer Science, vol. 173, Springer-Verlag, Berlin, 1984, pp. 177-195.

[5] Bresciani P., Miglioli P., Moscato U., Ornaghi M. - PAP: Proofs as Programs - (abstract), JSL, Vol. 51, n.3, 1986, pp. 852-853.

[6] Broy M., Wirsing M. - On the algebraic extension of abstract data types - in: Diaz J., Ramos I. (ed.) - Formalization of programming concepts - Lecture Notes in Comp. Sci. vol. 107, Springer-Verlag, Berlin, 1981.

[7] Chang C.C., Keisler H.J. - Model theory - North-Holland, 1973.

[8] Girard J. - The system F of variable types 15 years later - Report of CNRS, Paris, 1985.

[9] Goad C. - Computational uses of the manipulation of formal proofs - Rep. STAN-CS-80-819, Stanford University, 1980.

[10] 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.

[11] Goto S. - Program synthesis through Gödel's interpretation - Mathematical studies of information processing, (proceedings, Kyoto, 1978), Lecture Notes in Computer Science, vol.75, Springer-Verlag, Berlin, 1979, pp. 302-325.

[12] Guttag J., Horning J. - The algebraic specification of abstract data types - Acta Informatica 10, 27-52, 1978.

[13] Howard W.A. - The formulae-as-types notion of construction - in To Curry H.B.: essays on combinatory logic, lambda calculus and formalism, Academic Press, London, 1980.

[14] Kirk R.E. - A result on propositional logics having the disjunction property - Notre Dame Journal of Formal Logic, 23,1, 71-74, 1982.

[15] Kreisel G., Putnam H. - Eine unableitbarkeitsbeismethode für den intuitionistischen Aussagenkalkul - Archiv für Mathematische Logik und Grundlagenforschung, 3, 74-78, 1957.

[16] Martin-Löf P. - Constructive Mathematics and Computer Programming - Logic, Methodology and Philosophy of Science VI, L. Cohen, J. Los, H. Pfeiffer, K. Podewski (ed.), North-Holland, Amsterdam, 1982, pp.153-175.

[17] Miglioli P., Moscato U., Ornaghi M. - Constructive theories with abstract data types for program synthesis - Proceedings of the symposium Mathematical Logic and its Applications, Plenum Press, New York, 1988, pp.293-302.

[18] Miglioli P., Ornaghi M. - A logically justified model of computation I,II - Fundamenta Informaticae, IV.1,2, 1981.

[19] Nordstrom B., Smith J.M. - Propositions, Types and Specifications of Programs in Martin-Löf's Type Theory - BIT, Vol. 24, n.3, 1984, pp.288-301.

[20] Troelstra A.S. - Metamathematical investigation of Intuitionistic Arithmetic and Analysis - Lecture Notes in Mathematics, vol.344, Springer-Verlag, Berlin, 1973.

[21] Troelstra A.S. - Aspects of constructive mathematics - in: Barwise J. (ed.) - Handbook of Mathematical Logic, North Holland, Amsterdam 1977.

[22] Miglioli P., Moscato U., Ornaghi M. - PAP: a logic programming system based on a constructive logic - LNCS, n.306, Springer Verlag, 1988, pp.143-156.