Why Horn Formulas Matter in Computer Science: Initial Structures and Generic Examples
(extended abstract)

J.A. Makowsky

Department of Computer Science
Technion - Israel Institute of Technology
Haifa 32000, Israel
and
Institut für Informatik
Swiss Federal Institute of Technology
CH-8092 Zürich, Switzerland

Abstract: We introduce the notion of *generic examples* as a unifying principle for various phenomena in computer science such as *initial structures* in the area of abstract data types and *Armstrong relations* in the area of data bases. Generic examples are also useful in defining the semantics of logig programming, in the formal theory of program testing and in complexity theory. We characterize initial structures in terms of their generic properties and give a syntactic characterization of first order theories *admitting initial structures*. The latter can be used to explain why Horn formulas have gained a predominant role in various areas of computer science.

1. Introduction

Verification by example has always been alternative to formal deduction. Historically, in mathematics, it usually also preceeded the development of formal deduction methods. The Babylonians "knew" that $(x-y)^2=x^2+2xy+y^2$ but they did not have a notational system which allowed them carry out a formal, i.e. algebraic, proof. Instead they wrote $(3+5)^2=3^2+2x3x5+5^2$, from which they immediately concluded all the other instances of the general formula. The choice of the particular instance $x=3,y=5$ is important here. It is clear why $x=1,y=2$ would confuse the matter, and we informally describe an appropriate choice of an instance as the finding of a "generic" example. The art of finding "generic" examples has been pushed to the extreme in Euclidean plane geometry, where we convince ourselves of many theorems by just drawing *one* picture of a non-degenerate case. The generalization of this approach to other areas of reaoning is usually highly non-trivial. In algebraic geometry, for example, a satisfactory definition of "generic points" was only found in this century.

In computer science one is often concerned with the specification and analysis of algorithms and programs. Methods for formal specification and verification of programs have been developed intensively without leaving too much impact on the *practical* programmers. These methods are all very much in the spirit of formal deduction. The use of "generic" examples can be observed occasionally with various degrees of explicitness. Strassen [Str74] and his school have used the generic points of algebraic geometry with considerable success to obtain lower bounds in algebraic complexity theory. Recent work in the mathematical foundation of program testing, as presented in the survey edited by B. Chandrasekaran and S. Radicchi [CR81], focus around various notions of "generic" input. In data base theory, W. Armstrong [Ar74] has introducted a kind of "generic" relation for functional dependencies and R. Fagin has investigated the possibilities of generalizing this for implicational dependencies [Fa82]. Last but not least there is M. Zloof's approach to data base query languages where queries are specified by giving "generic" examples, an approach he most recently generalized to operate more complex systems in office automation [Zl82]. It is not surprising that *specification and verification by example* is more appealing to the computer engineer than formal deduction. A look at Euclidean geometry can be revealing again: People involved in surveying and drawing plans have, in general, very little use for formal deductions Euclidean style, but are very much aware of the role of the "generic" non-generate configurations.

The purpose of this paper is to introduce some variations of notions of "*genericity*" which arise in abstract specification of data structures, in relational data bases and in logic programming. What these three areas of computer science have in common, is the use of *first order logic* as its basic specification language. In each of these areas *Horn formulas* play an important role. In algebraic specification of abstract data structures one first used pure *equational logic* with the semantics of *initial structures* as a specification language (hence algebraic). Later one felt the need to extend this to *conditional equations*, which are *universal Horn formulas* without relation symbols. In relational data bases various specification languages where introduced, such as the arrow notation between finite sets of attribute names, to express functional and multivalued dependencies. It was soon realized by R. Fagin, C. Beeri and others, that implicational dependencies, which are Horn formulas without function symbols, could capture all the previously considered cases (cf. [Fa82]). In logic programming Horn formulas are used both as a specification and a programming language because, as R. Kowalski put it, the allow a *procedural* interpretation (cf. [Ko79]).

From a semantic point of view all these approaches can be described in a similar way: Instead of thinking of arbitrary models (= algebraic structures, relations or first order structures resp.) one only considers a *restricted* class of structures (= initial structures, Armstrong relations, minimal Herbrand universes). These restricted classes have all in common some sort of genericity, $\exists^+$-*genericity*, which we shall describe below. Intuitively $\exists^+$-genericity captures rather well the notion of a *generic example*, here of a $\exists^+$-generic structure satisfying the required specifications.

Various attempts exist in the literature to explain why Horn formulas are the *right* class of formulas to be used in the respective context. B. Mahr and J. Makowsky [MM83] prove that under certain assumptions for the semantics of algebraic specifications, conditional equations form the largest specification language satisfying these assumptions. J. Makowsky and M. Vardi [MV84] characterize various classes of data base dependencies in terms of preservation properties under operations on relations which come from data manipulation. In logic programming,, it was shown by Tarnlund [Tarn77] that Horn logic is enough to program every recursive function, a result, stated in slightly different form in a different context, proven already by S. Aanderaa and independently by E. Börger. For an excellent survey see [Bö84].

Our main result in this paper is a characterization of Horn formulas in terms of the existence of $\exists^+$-generic structures. It simultaneously extends and unifies results of [MM83], [Ma84] and [VM84] and remedies objections raised to [MM83] by A. Tarlecki. It states that a first order theory T admits initial (= $\exists^+$-generic) models iff there is a set of definable partial functions such that adding functions to the vocabulary of T gives us a theory T_1, which is equivalent to a universal Horn theory. Additionally, if T is finite, this set of definable partial functions can be chosen to be finite, too. In other words, if we want to define a semantic over $\exists^+$-generic structures only, we can, without loss of generality, confine our specification language to universal first order Horn formulas. Without loss of generality can mean two things: Given a specification which is not a set of universal Horn formulas, then either we can find an equivalent set of universal Horn formulas or we have chosen the basic vocabulary (set of basic symbols) wrongly, and then the theorem tells us that there is a unambiguous way to correct this.

In detail the paper is organized as follows:

In section 2 we introduce the concepts of A-genericity and $\exists^+$-genericity and relate these definitions to initiality. We state the basic definability theorem for initial models; we characterize initial term models as A-generic models and initial models as $\exists^+$-generic pseudo-term models.
In section 3 we characterize first order theories which admit initial term models as the universal Horn theories. This theorem was already proved in [MM83].
In section 4 we establish the intersection property of first order theories admitting $\exists^+$-generic structures and review some classical model theoretic results on first order theories with the intersection property. From this we get that theories admitting $\exists^+$-generic models can always be axiomatized by universal-existential sentences. We also state a theorem of M. Rabin [Ra60], which characterizes first order theories with the intersection property.
In section 5 we state an analogue of Rabin's theorem to obtain our main result. We show that a first order theory admits initial models iff it is a partially functional $\forall\exists$-Horn theory.
No proofs are presented in this extended abstract, since the editors of these proceedings put a severe space limit on the papers to be presented. We hope that the complete paper will appear soon elsewhere.
In section 6 we state some conclusions.

The reader familiar with the introduction to model theory by G. Kreisel and J.L. Krivine will realize how much, in spirit, this work is influenced by chapter 6 of [KK66]. I am indebted also to A. Tarlecki for his remarks in our correspondence concerning [MM83], to S. Shelah, who suggested theorem 2.10 and to B. Mahr for the discussions around [MM83].

2. Initial models and genericity

In this section we deal with first order languages *with equality*. *Vocabularies (= similarity types)* are allowed to be *many-sorted* and may include *function symbols*, *relation symbols* and *constant symbols*. Vocabularies are denoted by τ,σ. A *τ-structure* $\mathbf{A}$ is a collection of *universes* (= sets) $A_1,...,A_n$, for each sort in τ one, together with interpretations for all the function, relation and constant symbols in τ. *τ-terms*, *atomic formulas* and *τ-formulas* are defined as usual. If T is a set of τ-formulas, φ is a τ-formula and $\mathbf{A}$ is a τ-structure we write $\mathbf{A}\models T$ if the universal closure of all the formulas φ in T are true in $\mathbf{A}$. We write $T\models\varphi$ if in every τ-structure $\mathbf{A}$ such that $\mathbf{A}\models T$ we also have that $\mathbf{A}\models\varphi$. We call sets of τ-formulas *theories* and formulas *without free variables* also τ-sentences. We call τ-structures also *models* and

denote by *Mod(T)* the class of t-structures **A** such that $\mathbf{A} \models T$.

2.1. Defintions:

(i) Let K be a class of τ-structures closed under isomorphisms and $\mathbf{A} \in \mathbf{K}$. We say that **A** *is initial in* **K** *(is an initial model for* **K***)* if for every structure $\mathbf{B} \in \mathbf{K}$ there is a *unique* homomorphism $h_B: \mathbf{A} \to \mathbf{B}$.

(ii) If **K** is of the form *Mod(T)*, where T is some first order theory, we also say that **A** is *initial for T.*

(iii) A τ-structure **A** is a *term model (reachable model)* if for every $a \in \mathbf{A}$ there is τ-term t such that its interpretation $\mathbf{A}(t)$ in **A** is the element a.

Next we introduce the concept of *generic* structures for first order theories and relate it to initial structures.x

2.2. Definitions: Let **K** be a class of τ-structures closed under isomorphisms and $\mathbf{A} \in \mathbf{K}$. Let Σ be a set of first order sentences (i.e. formulas without free variables).

(i) We say that **A** *is generic* in **K** *for* Σ if *for* every $\varphi \in \Sigma$ we have that $\mathbf{A} \models \varphi$ iff $\mathbf{B} \models \varphi$ for every $\mathbf{B} \in \mathbf{K}$.

(ii) If Σ is the set of atomic τ-formulas we say *A-generic* instead of generic for Σ.

(iii) Let $\exists^+$ be the set of τ-formulas of the form $\exists x \bigwedge^n \varphi_i$ with each φ_i an atomic formula, $x=(x_1,x_2,...,x_n)$; and $\exists$ be the set of τ-formulas of the form $\exists x \psi(x)$ with ψ quantifier free.

(iv) If Σ is the set of $\exists^+$-sentences we say $\exists^+$*-generic* instead of generic for Σ.

2.3. Remarks:

(i) If $\Sigma_0 \subset \Sigma$ and **A** is Σ-generic then **A** is also Σ_0-generic.

(ii) If **A** is a *A*-generic term model then **A** is an initial term model.

2.4. Theorem ($\exists^+$-genericity):

Let **K** be a class of τ-structures closed under isomorphisms and $\mathbf{A}_I$ be initial for **K**. Then $\mathbf{A}_I$ is $\exists^+$-generic.

2.5. Corollary: Let **K** be a class of τ-structures closed under isomorphisms and $\mathbf{A} \in \mathbf{K}$. Then **A** is an initial term model for **K** iff **A** is *A*-generic.

2.6. Definitions: Let T be a set of τ-sentences, let $\mathbf{A}$ be a τ-structure and $a \in A$ be an element of the universe of $\mathbf{A}$.

(i) a is *definable over* $\mathbf{A}$ if there is a τ-formula $\varphi_a(x)$ with x the only free variable of φ such that $\mathbf{A} \models \varphi_a(a)$ and if $\mathbf{A} \models \varphi_a(b)$ for any $b \in A$ then $\mathbf{A} \models a = b$. We call φ_a the defining formula of a.

(ii) a is $\exists^+$*-definable* ($\exists$*-definable, atomically definable*) over $\mathbf{A}$ if a is definable over $\mathbf{A}$ and the defining formula is an $\exists^+$-formula ($\exists$-formula, conjunction of atomic formulas).

(iii) a is *definable over* T if there is a τ-formula $\varphi_a(x)$ with x the only free variable of φ_a such that $\mathbf{A} \models \varphi_a(a)$ and $T \models \forall x \forall y (\varphi_a(x) \wedge \varphi_a(y) \Rightarrow x = y)$.

(iv) a is $\exists^+$*-definable* ($\exists$*-definable, atomically definable*) over T if a is definable over T and the defining formula is an $\exists^+$-formula ($\exists$-formula, conjunction of atomic formulas).

(v) We say that $\mathbf{A} \models T$ is a *pseudo term model of* T if every element $a \in A$ is $\exists^+$-definable.

The following theorem shows that in an initial model of a theory T is always a pseudo term model.

2.7. Theorem ($\exists^+$-definability): Let T be a first order theory and let $\mathbf{A}_I$ be an initial model of T. Then every $a \in A$ is definable over T by a $\exists^+$-formula φ_a. In other words $\mathbf{A}_I$ is a pseudo term model.

We now are in a position to characterize initial models as pseudo term models which are $\exists^+$-generic.

2.8. Theorem: Let T be a first order theory and let $\mathbf{A}$ be a model of T. Then $\mathbf{A}$ is initial (for T) iff $\mathbf{A}$ is a $\exists^+$-generic pseudo-term model.

3. Characterizing first order theories which admit initial term models

In this section we characterize first order theories which admit initial term models. Such a characterization was first given in [MM83], based on a theorem due to Mal'cev [Mal56]. In [Mal56] there is a minor mistake as pointed out by [Mo59], which propagated inot [MM83] in as far as one had to assume that every first order theory admitting initial

/.3.1. Multiple physical clocks

The objective is also to obtain a unique physical time frame within the system so that consistent schedules may be derived from a total chronological ordering of actions occurring in the system. When several clocks are used it is not enough for the clocks individually to run at approximately the same rate. They must be kept synchronized so that the relative drifting of any two clocks is kept smaller than a predictable constant. In [Lamport78] a solution to accomplish this is given. The system under consideration is modelled after a strongly connected graph of processes with diameter d. Every process is provided with a clock and every t, a synchronization (sync) message is sent over every arc. A sync message contains a physical timestamp T. Upon receiving a sync message, if needed, a process should set forward its local clock to be later than the timestamp value contained in the incoming message. It is assumed that both a lower bound u and an upper bound u + z are known for interprocess message transit delays. Let k be the intrinsic accuracy of each clock (e.g. $k < 10^{-6}$) and e the allowed drifting of any two clocks. If $e/(1-k) <= u$ and $e << t$ then it is possible to compute the approximate value of e which is $d(2kt + z)$.

Depending on the requirements as regards clocks' relative drifting and the validity of the assumptions as regards transit delay boundaries, one may decide:

* either to take the risk of missing some sync messages from time to time, because of some excessively large message transit delays, thus achieving what could be called probabilistic synchronization.

* or not to take this risk. Then, if the upper

bound chosen for message transit delays has to be rather large, one should evaluate the consequences as regards performance. The key parameter here is the ratio z/u. For example in Arpanet, z and u are in the order of several hundreds of milliseconds.

The use of timestamps to obtain orderings of actions in a distributed system was suggested first in [Thomas76].

7.3.2. Multiple logical clocks

A logical clock as described in [Lamport78] should be viewed as a function C which assigns a number to any action initiated locally. Such logical clocks may be implemented by counters. In a system where each producer owns a logical clock, the problem is to guarantee that the system of clocks satisfies some condition F so that a particular ordering may be built on the set of actions initiated by producers. For example, using the irreflexive partial ordering introduced in section 3, condition F would be as follows: for any actions a (in i) and b (in j), if a -> b then C(i,a) < C(j,b). In order to meet condition F, the following rules must be obeyed by producers.

Rule 1: each producer i increments C(i) between any two successive actions.

Rule 2: if action a is the sending of a message m by producer i, then the message m contains a timestamp T(m) = C(i,a). Upon receiving a message m, producer j sets C(j) greater than or equal to its present value and greater than T(m).

Any system of logical clocks satisfying condition F can be used to place a total ordering, denoted by <<, on any set of actions. It is only necessary to use any arbitrary total ordering < of the producers (e.g. by using their names). The ordering << is defined as follows: a << b if and only if either C(i,a) < C(j,b) or C(i,a) = C(j,b) and i < j. The synchronization mechanism defined by rules 1 and 2 and the total ordering << allow for the building of consistent schedules of actions. The ordering << is not unique and it may not be equivalent to a chronological ordering. This is why it may be necessary to implement such a system of logical clocks on a system of several physical clocks (see previous section).

General comments

Synchronization mechanisms built out of several physical or logical clocks have the common characteristic that they are not based on mutual exclusion. This may be particularly advantageous in distributed systems.

7.3.3. Utilization of a circulating privilege

Synchronization mechanisms may take advantage of the fact that producers are given unique and permanent names. This defines a total ordering on the set of producers. Such an ordering may be used to view producers as being organized as on a chain or as on a loop. Each producer has a unique predecessor and a unique successor. Such a logical structuring does not imply any particular physical topology.

Pair-wise shared variables: A synchronization mechanism based on the concept of a logical ring has been presented in [Dijkstra74]. Possession of a control privilege may be inferred by every producer from the observation of a variable shared with one of its two neighbours.

The next theorem, due to Rabin [Rab60], characterizes theories with the Intersection Property. For this let $\varphi(x,y)$ be a first order formula with free variables $x=x_1,x_2,...,x_t$ and $y=y_1,y_2,...y_n$ and let k be a natural number. We denote by $N(k,y,\varphi(x,y))$ the first order formula which says that there are exactly k different n-tuples satisfying the formula $\varphi(x,y)$.

4.7. Theorem: (Rabin [Rab60]) A necessary and sufficient condition for a first order theory T to have the Intersection Property is that for every $\forall\exists$-sentence $\forall x\exists y\varphi(x,y)$ which is a consequence of T, there exist two sequences of quantifier-free formulas

$$\sigma_1(x,u),\sigma_2(x,u),...,\sigma_\mu(x,u) \quad \text{and} \quad \theta_1(x,y,z),\theta_2(x,y,z),.....,\theta_\mu(x,y,z)$$

and a sequence of natural numbers $k_1,k_2,...,k_\mu$ such that

$$T\models\forall x(\forall u\sigma_1(x,u)\vee\forall u\sigma_2(x,u)\vee...\vee\forall u\sigma_\mu(x,u))$$

and for $1\leq i\leq\mu$

$$T\models\forall x(\forall u\sigma_i(x,u)\Rightarrow N(k_i,y\exists z\varphi(x,y)\wedge\theta_i(x,y,z))).$$

In the next section we want to give a similar characterization for first order theories admitting initial models. Our next goal is to show the existence of initial models for certain theories which have the Intersection Property and which are preserved under products. For this we need some more definitions.

4.8. Definitions: Let T be a first order theory with the Intersection Property. A model $\mathbf{A}_0$ of T is a *core model* if there is no proper submodel $\mathbf{B}\subset\mathbf{A}_0$ such that $\mathbf{B}\models T$. If $\mathbf{A}$ is a model of T and $\mathbf{A}_0\subset\mathbf{A}$, $\mathbf{A}_0\models T$ is a core model we say that $\mathbf{A}_0$ is a *T-core of* $\mathbf{A}$.

4.9. Lemma: Let T be a first order theory with the Intersection Property. Then every model $\mathbf{A}$ of T has a T-core $\mathbf{A}_0$.

4.10. Proposition: Let T be a first order theory with the Intersection Property. Then every core model of T is an $\exists$-term model.

4.11. Definition: A first order theory T is *pseudo algebraic* if T is preserved under products, has the Intersection Property and if every core model of T is a pseudo term model.

4.12. Theorem: A pseudo algebraic first order theory T has an initial model $\mathbf{A}_I$.

A converse of theorem 4.12 will proved in the next section.

5. Charcterizing first order theories which admit initial models

The purpose of this section is to characterize first order theories which admit initial models. We first want to show that such a theory is equivalent to an $\forall\exists$-Horn theory.

5.1. Theorem: Let T be a first order theory which admits initial models. Then:
(i) T is equivalent to an $\forall\exists$-Horn theory $T_{\forall\exists H}$.
(ii) If T is finite, so is $T_{\forall\exists H}$.

Next we want to state an analogue of Rabin's theorem (theorem 4.7) for theories which admit initial models.

5.2. Theorem: Let T be a first order theory which admits initial models. Then for every $\forall\exists$-sentence $\forall x\exists y\varphi(xy)$ which is a consequence of T, there exist two sequences of formulas

$$\sigma_1(x,u),\sigma_2(x,u),...,\sigma_\mu(x,u) \quad \text{and} \quad \theta_1(x,y,z),\theta_2(x,y,z),.....,\theta_\mu(x,y,z),$$

where σ_i are quantifier free formulas and θ_i are $\exists^+$-formulas, such that

$$T\models\forall x(\forall u\sigma_1(x,u)\vee\forall u\sigma_2(x,u)\vee...\vee\forall u\sigma_\mu(x,u))$$

and for $1\leq i\leq\mu$

$$T\models\forall x(\forall u\sigma_i(x,u)\Rightarrow\exists!y(\exists z(\varphi(x,y)\wedge\theta_i(x,y,z)))$$

5.3. Definition: We call a first order theory which satisfies the conclusion of theorem 5.2

partially functional. This is justified since theorem 5.2 says that every $\forall\exists$-formula which is a consequence of T can be Skolemized with finitely many partial functions.

5.4. Corollary: Let T be a first order theory which admits initial models. Then every $\exists$-term model $\mathbf{A}$ is a pseudo term model.

We need another well known result from model theory, see e.g. ([CK73]):

5.5. Theorem: Let T be an $\forall\exists$-Horn theory. Then T is preserved under products.

Putting everything together we obtain:

5.6. Theorem: (Main theorem) Let T be a first order theory. The following are equivalent:
(i) T admits initial models;
(ii) T Is equivalent to a partially functional $\forall\exists$-Horn theory.
(iii) T is pseudo algebraic.

6. Conclusions

We have given a characterization of universal Horn theories in terms of the existence of initial, or equivalently, A-generic term models (theorem 3.9) and a characterization of partially functional $\forall\exists$-Horn theories in terms of the existence of initial, or equivalently $\exists^+$-generic pseudo term models (theorem 5.6). The latter essentially says that a first order theory which admits initial models which are not term models does so by oversight: The vocabulary (similarity type) was badly chosen, such as not to allow that all elements are denoted by some term. This can be almost remedied: Either by adding definable partial Skolem functions or by allowing pseudo terms, i.e. elements uniquely definable by $\exists^+$-formulas.

The paper also sheds more light into the question why in [ADJ75] initial structures were proposed as the framework for abstract data types. We have given in theorem 2.13 a characterization of initial structures as $\exists^+$-generic pseudo term models. For somebody not familiar with category theory this may be more appealing since it relates directly to or concept of verification by example. However, this characterization has also its technical

merits for it provides the missing link between the category theoretic concept and the model theoretic tools needed to prove 5.9.

Last but not least we have yet added another explanation as to why Horn formulas play such an important role in various branches of computer science. We have shown that universal Horn theories (partially functional $\forall\exists$-theories) are exactly the framework in which the notion of a generic example can be applied. This should prevent other researchers from trying to generalize Logic Programming or the semantics abstract data types to larger classes of first order formulas. If it has to be generalized then the direction chosen by R.M. Burstall and J.A. Goguen in [GB84] seems to be much more appropriate.

References

[ADJ75] Goguen, J.A., Thatcher, J.W., Wagner, E.G. and Wright, J.A.; Abstract data types as initial algebras and the correctness of data representations, Proc. of Conf. on Computer Graphics, Pattern Recognition and Data Structures, 1975, pp. 89-93.

[Ar74] Armstrong, W.W.; Dependency structures of database relationships Proc. IFIP 74, North Holland 1974, pp. 580-583.

[Bo84] Börger, E.; Decision problems in predicate logic, in: Logic Colloquium '82, G. Lolli, G. Longo and Am Marcja eds., North Holland 1984, pp. 263-302.

[CK73] Chang, C.C. and Keisler, H.J.; *Model Theory*, North Holland 1973.

[CR81] Chandrasekaran, B. and Radicchi, S.; *Computer Program Testing*, North Holland 1981.

[Fa82] Fagin, R.; Horn Clauses and data base dependencies, J.ACM vol. 29 (1982) pp. 252-285.

[GB84] Goguen, J.A. and Burstall, R.M., Introducing institutions, in Proc. of Logic Programming Workshop, E. Clarke ed., Lecture Notes of Computer Science 1984, to appear.

[KK66] Kreisel, G. and Krivine, J.L.; Elements de logique mathematique, Dunod 1966.

[Ko79] Kowalski,R.; *Logic for Problem solving*, North Holland 1979.

[McK43] McKinsey, J.C.C.; The decision problem for some classes of sentences without quantifiers, Journal of Smb. Logic vol. 8 (1943) pp. 61-76.

[Mal56] Mal'cev, A.; Quasi primitive classes of abstract algebras, in: The Metamathematics of algebraic systems, collected papers of A.I. Mal'cev, North Holland 1971, pp. 27-31.

[MM83] Mahr, B. and Makowsky, J.A.; Characterizing specification languages which admit initial semantics, Proc. of 8th CAAP, Springer LNCS vol. 159 (1983) pp. 300-316.

[Mv84] Makowsky, J.A. and Vardi, M.Y.; On the expressive power of data dependencies, submitted 1984.

[Ma84] Makowsky, J.A.; Model theoretic issues in Computer Science, Part I: Relational data bases and abstract data types, in: Logic Colloquium '82, G. Lolli, G. Longo and A. Marcja eds., North Holland 1984, pp. 303-344.

[Mo59] Mostowski, A.; Review of [Mal56], Journal of Symb. Logic vol. 24 (1959) p. 57.

[Ra60] Rabin, M.; Characterization of convex systems of axioms, Notices AMS, Abstract 571-65 (1960) p. 505.

[Ra62] Rabin M.; Classes of models and sets of sentences with the intersection property, Ann.Fac.Sci. Universite de Clermont, vol. 7.1 (1962) pp. 39-53.

[Rob63] Robinson, A.; *Introduction to model theory and the Metamathematics of algebra*, North Holland 1963.

[Sm84] Smorynski, C.; Lectures on non-standard models of arithmetic, in Logic Colloquium '82, Lolli, G. Longo, G. and Marcja, A. eds., North Holland 1984, pp. 1-70.

[St74] Strassen, V.; Polynomials with rational coefficients which are hard to compute, SIAM J.Comput. vol. 3.2 (1974) pp. 128-149.

[Ta52] Tarski, A.; Some notions and methods on the borderline of algebra and metamathematics, Proc.Int.Congr.of Mathe., Cambridge, MA vol. 1 (1952) pp. 705-720.

[Tar84] Tarlecki, A.; Free constructions in abstract algebraic institutions, draft February 1984.

[Tarn77] Tarnlund, S.A.; Horn clause computability, BIT vol. 17 (1977) pp. 215-226.

[Zi82] Zloof, M.M.; Office-by-example: A business language that unifies data and word processing and electronic mail, IBM Syst.J. vol. 21.3 (1982) pp. 272-304.