Revista Colombiana de Matemáticas
Volumen **31** *(1997), páginas 77–98*

A deductive calculus for conditional equational systems with built-in predicates as premises

Mauricio Ayala-Rincón
Universidade de Brasília, BRASIL

Abstract. Conditional equationally defined classes of many-sorted algebras, whose premises are conjunctions of (positive) equations and built-in predicates (constraints) in a basic first-order theory, are introduced. These classes are important in the field of algebraic specification because the combination of equational and built-in premises give rise to a type of clauses which is more expressive than purely conditional equations. A sound and complete deductive system is presented and algebraic aspects of these classes are investigated. In particular, the existence of free algebras is examined.

Key words and phrases. Algebraic specification, rewriting systems, theorem proving.

1991 AMS Mathematics Subject Classification. Primary 68Q65. Secondary 68Q42, 68T15, 03C05, 08A70.

1. Introduction

The need to use conditional equations appears initially in universal algebra in order to represent some algebraic structures, for example the left cancellative law can be expressed by the conditional equation $x * y = x * z \implies y = z$. Classes of algebras presented by equations and conditional equations are called varieties and quasivarieties (or positive equational universal Horn classes), respectively. The use of conditional equations has also been studied in the fields of abstract data types and algebraic specification because they provide easier and more elegant specifications having a greater expressive power than (unconditional) equations.

A combination of many-sorted (positive equational) universal Horn clauses with built-in predicates in a basic theory as premises is considered. The purpose of this combination is to improve the expressiveness of the (positive equational) universal Horn clauses in such a way that one can request for (not necessarily equational) predicates in a first-order theory over a specific signature. Built-in predicates are logical objects that are incorporated (as conditions) into the syntactical structure of (positive equational) universal Horn clauses. This gives rise to a class of conditional clauses with conditions formed by a built-in predicate and the usual conjunction of equations, of the following form:

$$u_1 = v_1 \wedge \cdots \wedge u_n = v_n \wedge C \implies u = v.$$

Universal Horn classes with built-in predicates are presented by this type of clauses. These combined classes are more expressive than quasivarieties since theories which cannot be axiomatized by universal Horn clauses can be incorporated as built-in predicates.

It is well-known that free algebras exist for varieties as well as for quasivarieties. In contrast, this is not a natural property of universal Horn classes with built-in predicates because, for instance, the built-in predicates can be negative equational conditions and in general negative conditional equationally defined classes have no free algebras. However, under certain restrictions which can be conceived as a type of initiality property (or restriction to the standard model) of the basic theory, the existence of free and even initial algebras is guaranteed.

Conditional rewriting has been suggested as the basic operational mechanism for deduction in universal Horn classes with built-in predicates [AR93]. Conditional rewriting systems have been used for equational deduction in conditional equationally defined classes of algebras. Sets of equations and conditional equations can be considered as functional programs with rewriting as its computation mechanism. Whereas predefined operations are available in common programming languages they are not naturally incorporated into rewriting techniques. Therefore it is of great interest to amalgamate built-in algorithms and rewriting.

This work is focused on the algebraic (not on the operational) properties of positive equational universal Horn classes with built-in predicates. Firstly, basic theories are defined. Afterwards, the syntax and semantics of specifications with combined premises (conjunctions of equations and predicates in a basic theory) are investigated. Finally, a deductive calculus for this class of specifications is introduced, its completeness proved and appropriate restrictions to guarantee the existence of free algebras are given.

Related work: Already O'Donnell considered arbitrary predicates as conditions of rules [O'D77]. Padawitz [Pad88] has presented in detail a calculus for universal Horn clauses from the algebraic point of view. His approach is more general than that presented here, because he does not separate between logic and algebraic concepts. Here this separation is necessary because in the present work the intention is to formalize an operational approach: the combination of decision procedures for evaluating many-sorted first-order logic predicates and deduction based on term rewriting techniques for manipulating the whole specification. With more detail than Padawitz (at least from the operational point of view), Kaplan [Kap83], [Kap84] has developed a calculus for equational universal Horn clauses but without the special kind of built-in predicates incorporated here. Kaplan's work is important principally because he was the first person developing with precision an operational approach to manipulate these restricted classes of universal Horn clauses by extending the well-known purely equational rewriting techniques. Vorobyov [Vor89] has treated deduction for conditional specifications whose conditions are built-in predicates combining rewriting and built-in algorithms. Kirchner, Kirchner and Rusinowitch [KKR90] have focused the deduction from the point of view of constraints. Neither Vorobyov nor the other authors admit standard equational conditions as it is done in this work. Efforts to combine built-in predicates (and their corresponding decision procedures) and equational conditions as premises of conditional specifications, were made in [AR93].

Dershowitz and Okada [DO90] studied conditional rewriting systems with (informal) built-in predicates and standard equational conditions as premises. They briefly treated termination, confluence and a critical pair criterion for their systems. Precise proofs of their suggested results on confluence can be found in [AR94]. Avenhaus and Becker [AB92] presented a method to integrate built-in operations that are described by a given built-in algebra into conditional rewriting. They do not allow new sorts in the extended signature and the variables are restricted to range over basic terms. Later on Becker [Bec94] presented an operationalization of clausal specifications where the axioms are positive/negative conditional equations admitting predefined algebras. The clauses are treated by rewriting. His approach can be considered as an attempt to provide a general framework to handle specifications with predefined algebras in a semantically clean way.

The principal differences between the present approach (and Vorobyov's approach) and the previous ones is that whereas here all models of the basic theory are considered, they work with a specific model of the basic theory giving restrictions and assumptions on the built-in part which allow an operational treatment by purely rewriting and validity check avoiding case-splittings. Their approaches are appropriate to compute in algebraic structures with built-in parts but they are not sufficient for equational deduction (neither in the whole theory nor in the initial theory).

An extended abstract of this work appears in [AR95].

2. Basic theories

Appropriate definitions of basic theories, whose objects will be further incorporated (as conditions) into the syntactical structure of universal Horn clauses, are given. Basic knowledge about algebraic specification (see [EM85],[Wec92]) and model theory is assumed (see [CK90], [Gal87]).

By adding to the formalism of first-order logic (with equality) the notion of sort, one obtains a flexible logic called many-sorted first-order logic (with equality) (see [Gal87]), which enjoys the same properties as first-order logic. The purpose of many-sorted first-order theories is to express properties of structures of different sorts which appear when one is interested in axiomatizing data structures found in computer science (lists of natural numbers, arrays of objects indexed by integers, etc.).

Definition 2.1. *An S_0-sorted signature is a set Σ_0, whose elements are called operation (or function) and predicate symbols, together with an arity function $\mathcal{AR} : \Sigma_0 \to S_0^* \times (S_0 \cup \{\lambda\})$, which assigns to each operation symbol f of Σ_0 an ordered pair (w, s) and to each predicate symbol p of Σ_0 an ordered pair (w, λ) with their first component w a word over S_0, called the arity or domain of f or p, respectively, and their second component s an element of S_0, called the sort or range of f, and λ meaning that p is a predicate symbol, respectively.*

The language of many-sorted first-order logic is given by an S_0-sorted signature Σ_0; a family of disjunct countably infinite sets V_s of variables for every sort $s \in S_0$ ($V_0 = \bigcup_{s \in S_0} V_s$); the logical connectives &, $\neg$ and the logical universal quantifier $\forall$[1]; one binary relation symbol $=$[2] and auxiliary symbols (,).

The notions of (well-formed) *term* and *formula*, *atomic formula*, *free* and *bound* occurrences of a variable in a formula, *universal* and *existential closure* of a formula *sentences* (i.e., formulae with no free variables) and of *quantifier-free* formulae (i.e., formulae with no bound variables) are assumed to be known. $t(x_1, \ldots, x_n)$ denotes a term t whose variables belong to $\{x_1, \ldots, x_n\}$. Similarly, $P(x_1, \ldots, x_n)$ denotes a formula whose free variables form a subset of $\{x_1, \ldots, x_n\}$.

[1] The others logical connectives (i.e., $\vee, \Rightarrow, \Leftrightarrow$) and the existential quantifier ($\exists$) can be seen as abbreviations of &, $\neg$ and $\forall$. For example, $\exists x P(x)$ is an abbreviation of $\neg\forall x \neg P(x)$.

[2] To be more precise, one should consider one binary relation symbol $=_s$ and one universal quantifier $\forall_s$ for each sort $s \in S$.

As usual, logical axioms and rules of inference (*modus ponens* and universal generalization) are needed to transform the above syntactical notions into a formal system.

For a set of sentences T and a formula P, $\vdash P$ and $T \vdash P$ mean that P is a *theorem* of Σ_0 and that there is a *proof* of P from the logical axioms and T, respectively. As the logical axioms are always assumed, it is said that there is a proof of P from T, or P is deducible from T, whenever $T \vdash P$.

Definition 2.2. *A set T of sentences of Σ_0 is said to be inconsistent iff every formula of Σ_0 can be deduced from T. Otherwise T is consistent. A sentence P is consistent iff $\{P\}$ is. A sentence P is said to be T-consistent iff $T \cup \{P\}$ is. A quantifier-free formula $P(x_1, \dots, x_n)$ is said to be T-consistent iff its existential closure $\exists x_1, \dots, x_n(P(x_1, \dots, x_n))$ is T-consistent.*

Definition 2.3. *An interpretation $\mathfrak{A}$ for Σ_0 consists of an S_0-sorted universe $A_0 = \bigcup_{s \in S_0} A_s$, where $A_s \neq \varnothing$ for all $s \in S_0$. In this universe, each predicate symbol $p : w$ and each operation symbol $f : w \to s$ in Σ_0 correspond to a w-placed relation $p^{\mathfrak{A}} : A^w$ and to a function $f^{\mathfrak{A}} : A^w \to A_s$, respectively. Nullary operation symbols $f : \to s$ correspond to constants.*

As usual the notation $\mathfrak{A} \models P$ is used to denote that the sentence P is *valid* in the interpretation $\mathfrak{A}$ or, equivalently, that $\mathfrak{A}$ is a model of P.

Definition 2.4. *Given a set of sentences T. $\mathfrak{A}$ is said to be a model of T iff $\mathfrak{A}$ is a model of each P in T. As usual, this is denoted by $\mathfrak{A} \models T$. A sentence or a set of sentences is satisfiable iff it has at least one model. A sentence is a logical consequence of a set of sentences T, in symbols $T \models P$, iff every model of T is a model of P. A set of axioms of a theory T_0 is a set of sentences with the same logical consequences as T_0.*

Remark 2.5. Although many-sorted first-order logic is very convenient, it is not an essential extension of standard one-sorted first-order logic in the sense that there is a translation of many-sorted logic into one-sorted logic. For details see [Gal87]. In this work, results of the classical one-sorted first-order logic are indiscriminately applied to the many-sorted first-order logic. $\lhd$

A basic theory is defined as follows.

Definition 2.6. *A basic theory T_0 of Σ_0 is an arbitrary (consistent) many-sorted first-order theory with equality.*

The Presburger arithmetic ($\mathcal{PAR}$), the additive number theory ($\mathcal{ANT}$) and the successor arithmetic ($\mathcal{SA}$) are examples of (one-sorted) basic theories. The language of the successor arithmetic consists of the constant 0, the unary function symbol s for successor and binary relation $\leq$. $\mathcal{SA}$ is the set of all first-order formulae from this language valid in the standard model of natural numbers, where all symbols get their usual meanings. The Presburger arithmetic (or theory of integers under addition) was originally defined over the domain of integers with the usual constants 0 and 1 and with arithmetical equality and addition only. It is well-known that the Presburger arithmetic is not finitely axiomatizable but it was showed that it is decidable and complete, by the method of elimination of quantifiers [Pre29] (see for example [Coo72] for a relatively efficient algorithm using this method). Presburger's proof was extended later to include all the usual arithmetical relations $(<, >, \leq, \geq, =)$ over the domain of integers.

In this work Presburger formulae will be considered as those that can be built up from integer constants, integer variables, addition, the usual arithmetical relation "$<$", the first-order logical connectives and quantification. Formulae take their usual meaning. For example, the formula $\forall x \exists y (2y + x < 3 \Rightarrow x < y)$ falls within this class. Note that nx is an abbreviation for repeated addition but arbitrary multiplication is not permitted. The other arithmetical relations may easily be added; for example, $x = y$ can be defined as $x < y+1 \;\&\; y < x+1$ and $x \geq y$ as $y < x + 1$.

A unquantified Presburger formulae is a Presburger formulae having no quantifiers. The unquantified Presburger arithmetic, consisting of those unquantified Presburger formulae in the Presburger arithmetic, is decidable as subclass of the Presburger arithmetic. In particular, this subclass of formulae may be decided by Bledsoe's SUP-INF method; see [Sho81, ARG97].

3. Universal Horn clauses with built-in predicates

Built-in predicates are logical objects that are incorporated (as conditions) into the syntactical structure of universal Horn clauses. This results in a class of conditional clauses with conditions formed by the usual conjunction of equations and a built-in predicate.

Definition 3.1. *Built-in predicates are quantifier-free formulae of a basic theory.*

Remark 3.2. The unquantified $\mathcal{SA}$, $\mathcal{PAR}$ and $\mathcal{ANT}$ are decidable as subclasses of decidable theories. To consider only quantifier-free formulae appears to be very restrictive, however this is not the case because many existential quantified formulae can be abbreviated to new built-in predicate symbols; for instance, the formula $\exists x(x + x = y)$, in the Presburger arithmetic, can be conceived as the predicate $Even(y)$.

This restriction is made to incorporate the built-in predicates as conditions into the universal Horn clauses with one general universal quantifier for the whole clause. $\triangleleft$

In order to incorporate the built-in predicates as conditions into the structure of universal Horn clauses (defined in an S-sorted signature Σ), built-in objects are described in a built-in language given by an S_0-sorted signature Σ_0, as mentioned above, with the following restrictions:

Restriction 3.3 (on the signatures)**.** *Let Σ_0 be an S_0-sorted signature, the built-in language of a basic theory T_0. The introduction of new "syntactical" function symbols is captured as usual by the notion of a signature extension. Let S_0 and S be two sets of sorts such that $S_0 \subseteq S$. Let Σ_0 and Σ be S_0- and S-sorted signatures, respectively. Suppose that $\Sigma \setminus \Sigma_0$ does not contain function symbols with sort in S_0. Terms with sort in S_0 are called basic terms and in $S \setminus S_0$ extended terms.*

The previous restriction separates built-in from extended terms in such a way that the conservativeness of the extension is guaranteed.

The following example illustrates a signature with the Presburger arithmetic as built-in theory.

Example 3.4. Let $S_0 = \{int\}$, $S = S_0 \cup \{array, element\}$ be sets of sorts and Σ_0 and Σ be S_0- and S-sorted signatures, respectively, with the following declarations:

$$\begin{array}{lll}
\Sigma_0 : & \dots, -1, 0, 1, \dots & : \ \to int \\
 & + & : \ int \times int \to int \\
 & < & : \ int \times int \\
\Sigma : & \langle _, _, _ \rangle & : \ array \times int \times element \to array \\
 & _[_] & : \ array \times int \to element
\end{array}$$

Note that the predicates of the basic signature Σ_0 have no range. However they have a logical interpretation that will be syntactically embedded into the extended specification. Constants $\dots, -1, 0, 1, \dots$ are given because it will be assumed that the built-in language is sufficiently expressive to give a ground term for any element of the models of interest. This can be made without mentioning explicitly all integer constants. $\triangleleft$

Definition 3.5. *Let S_0, S, Σ_0, Σ and T_0 be restricted as before. (Σ, Σ_0, T_0) is said to be a signature with built-in theory T_0 over Σ_0.*

Definition 3.6. *Let (Σ, Σ_0, T_0) be a signature with built-in theory T_0 over Σ_0. A Σ-algebra over the built-in theory T_0,* A *consists of a model $\mathfrak{A}$ of T_0 with universe A_0, an S-sorted set A, whose restriction to S_0 is precisely A_0, and of* *a family of operations such that every operation symbol $f : w \to s$ of $\Sigma \setminus \Sigma_0$ is realized as an operation $f^{\mathrm{A}} : A^w \to A_s$, every operation symbol $f : w \to s$ of Σ_0 as $f^{\mathfrak{A}}$ and every predicate symbol $p : w$ of Σ_0 as $p^{\mathfrak{A}}$. When Σ is unspecified or unemphasized, a Σ-algebra over T_0 is simply said to be an algebra over T_0.*

Observe that this notion of Σ-algebra embodies an interpretation of the basic signature and even a model of T_0. This is a slight generalization of the usual notion of algebra which is well suited for the actual purposes.

In the sequel, when a Σ-algebra A is mentioned it will be assumed that $A, \mathfrak{A}$ and A_0 are its carrier set, its basic model and basic universe, respectively. The same for $\mathrm{B}, B, \mathfrak{B}$ and B_0.

Definitions of algebraic notions for Σ-algebras over built-in theories (viz subalgebra, homomorphism, direct product, direct limit, etc.) are not evident. In particular, note that one cannot straightforwardly adapt the definition of direct product for Σ-algebras over built-in theories because the interpretation of predicates is not obvious. For example, let $\mathfrak{A}$ be a model of $\mathcal{SA}$ and consider the direct product $A_0 \times A_0$ of its carrier set. The predicate $\leq$ cannot be interpreted for pairs in this direct product of sets; for instance, an interpretation for $(0, s(0)) \leq (s(0), 0)$ is not evident. The notion of subalgebra can be adapted for Σ-algebras over built-in theories by restricting the basic mod el of the algebra to remain identical in every subalgebra. The notion of homomorphism can also be defined by appropriately combining the logical and the algebraic notions as follows.

Definition 3.7. *Let $\mathfrak{A}, \mathfrak{B}$ be models of a basic theory, T_0 over Σ_0, with universes A and B respectively. $\mathfrak{B}$ is said to be homomorphic to $\mathfrak{A}$ if and only if there is an S_0-sorted surjective mapping $h : A \to B$ satisfying the following conditions:*

- *For each predicate symbol $p : w$ in Σ_0 and respective relations $p^{\mathfrak{A}} \subseteq A^w$ and $p^{\mathfrak{B}} \subseteq B^w$ and all n-tuples $(x_1, \ldots, x_n) \in A^w$, if $p^{\mathfrak{A}}(x_1, \ldots, x_n)$ then $p^{\mathfrak{B}}(h(x_1), \ldots, h(x_n))$.*
- *For each operation symbol $f : w \to s$ in Σ_0 and respective functions $f^{\mathfrak{A}} : A^w \to A_s$ and $f^{\mathfrak{B}} : B^w \to B_s$ and for all n-tuples $(x_1, \ldots, x_n) \in A^w$, $h(f^{\mathfrak{A}}(x_1, \ldots, x_n)) = f^{\mathfrak{B}}(h(x_1), \ldots, h(x_n))$.*

A mapping h satisfying the above is called a homomorphism of $\mathfrak{A}$ onto $\mathfrak{B}$. An isomorphism between $\mathfrak{A}$ and $\mathfrak{B}$ is a bijective homomorphism of $\mathfrak{A}$ onto $\mathfrak{B}$, such that for each predicate symbol $p : w$ in Σ_0 and respective relations $p^{\mathfrak{A}} \subseteq A^w$ and $p^{\mathfrak{B}} \subseteq B^w$ and all n-tuples $(x_1, \ldots, x_n) \in A^w$, $p^{\mathfrak{A}}(x_1, \ldots, x_n)$ iff $p^{\mathfrak{B}}(h(x_1), \ldots, h(x_n))$.

Observe that in the above definition, $\mathfrak{A}$ and $\mathfrak{B}$ are both assumed to be models of a basic theory T_0. Without this assumption, the existence of a homomorphism of a model $\mathfrak{A}$ of a theory T_0 onto $\mathfrak{B}$ does not imply that $\mathfrak{B}$ is a model of T_0. This is proved, for instance, by the fact that quasivarieties are not closed under homomorphic images. On the other hand, if $\mathfrak{A}$ is a model of a first-order theory T_0 and there is an isomorphism between $\mathfrak{A}$ and $\mathfrak{B}$ then $\mathfrak{B}$ is also a model of T_0. This follows from the notion of *elementary equivalence* and Lyndon homomorphism theorem; see [CK90].

Definition 3.8. *Let (Σ, Σ_0, T_0) be a signature and let* A *and* B *be Σ-algebras. An S-sorted mapping $h : A \to B$, whose restriction to A_0 is a homomorphism of $\mathfrak{A}$ onto $\mathfrak{B}$, is called a Σ-homomorphism if $h(f^{\mathrm{A}}(x_1, \ldots, x_n)) = f^{\mathrm{B}}(h(x_1), \ldots, h(x_n))$ for all operation symbols $f : w \to s$ in $\Sigma \setminus \Sigma_0$ and all n-tuples $(x_1, \ldots, x_n) \in A^w$.*

An S-sorted set of variables $V = \bigcup_{s \in S} V_s$ is fixed. Each V_s is a countably infinite set of variables of sort s. As before, V_0 denotes the set $\bigcup_{s \in S_0} V_s$ of basic variables.

Definition 3.9. *The S-sorted set of all Σ-terms over V, denoted by $T_\Sigma(V)$ is defined as follows: $T_\Sigma(V) = (T_\Sigma(V)_s | s \in S)$, where each set $T_\Sigma(V)_s$ of Σ-terms over V of sort s is recursively defined as the least set of words over $\Sigma \cup V_s$ such that $V_s \subseteq T_\Sigma(V)_s$ and, for all operation symbol $f : w \to s$ in Σ and all tuples $(t_1, \ldots, t_n) \in (T_\Sigma(V))^w$, $f(t_1, \ldots, t_n)$ belongs to $T_\Sigma(V)_s$.*

Note that for all predicate symbols $p : w$ in Σ_0, and all tuples $(t_1, \ldots, t_n) \in (T_\Sigma(V))^w$, $p(t_1, \ldots, t_n)$ expresses the predicate symbol applied to such a tuple (but it does not belong to $T_\Sigma(V)$). Moreover, observe that with all predicate symbols p in Σ_0, the logical connectives, universal quantifier, equality, etc. and the terms in $T_\Sigma(V)$ (or, more precisely, in $T_{\Sigma_0}(V_0)$), all possible basic formulae can be constructed.

An interpretation in $T_\Sigma(V)$ of all terms (in $T_\Sigma(V)$) and all possible predicate expressions (formed from predicate symbols $p : w$ in Σ_0 and tuples $(t_1, \ldots, t_n)$ in $(T_\Sigma(V))^w$) is given as follows.

Definition 3.10. *All terms t in $T_\Sigma(V)$ and predicate expressions formed from a predicate symbol $p : w \in \Sigma_0$ and a tuple $(t_1, \ldots, t_n) \in (T_\Sigma(V))^w$ have the following interpretation:*

- *if $t \equiv x \in V$ then $x^{T_\Sigma(V)} = x$;*
- *if $t \equiv f(t_1, \ldots, t_n)$ for some function symbol $f : w \to s$ in Σ and $(t_1, \ldots, t_n) \in (T_\Sigma(V))^w$ then $f^{T_\Sigma(V)}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$;*
- *$p^{T_\Sigma(V)}(t_1, \ldots, t_n)$ (or simply, $p(t_1, \ldots, t_n)$) iff $T_0 \models p(t_1, \ldots, t_n)$.*

In the sequel, when $T_\Sigma(V)$ is mentioned, it is supposed that all terms and predicate expressions are interpreted as above. It can be seen that $T_\Sigma(V)$

restricted to basic terms is an interpretation for the built-in language Σ_0. $T_\Sigma(V)$ is called the term Σ-interpretation for Σ_0.

Definition 3.11. *A congruence on $T_\Sigma(V)$ is an S-sorted equivalence relation R compatible with operation and predicate symbols: for all operation symbols $f : w \to s$ in Σ, if $(a_1, \dots, a_n)$, $(b_1, \dots, b_n) \in (T_\Sigma(V))^w$ and $(a_i, b_i) \in R$, for all $i = 1, \dots, n$, then R includes $(f^{T_\Sigma(V)}(a_1, \dots, a_n),\ f^{T_\Sigma(V)}(b_1, \dots, b_n))$ and for all predicate symbols $p : w$ in Σ_0, if $(a_1, \dots, a_n), (b_1, \dots, b_n) \in (T_\Sigma(V))^w$, $(a_i, b_i) \in R$, for all $i = 1, \dots, n$, and $p^{T_\Sigma(V)}(a_1, \dots, a_n)$ then $p^{T_\Sigma(V)}(b_1, \dots, b_n)$.*

The quotient $T_\Sigma(V)/R$ is defined as follows:

- *the carrier of $T_\Sigma(V)/R$ is the set of equivalence classes*
$$\{[a] = \{b | (b, a) \in R\} \mid a \in T_\Sigma(V)\},$$
denoted also by $T_\Sigma(V)/R$;
- *$\forall P : w \in \Sigma_0$, $\forall (a_1, \dots, a_n) \in (T_\Sigma(V))^w$, $P^{T_\Sigma(V)/R}([a_1], \dots, [a_n])$ whenever $P^{T_\Sigma(V)}(a_1, \dots, a_n)$;*
- *$\forall f : w \to s \in \Sigma$, $\forall (a_1, \dots, a_n) \in (T_\Sigma(V))^w$, $f^{T_\Sigma(V)/R}([a_1], \dots, [a_n]) = [f^{T_\Sigma(V)}(a_1, \dots, a_n)]$.*

T_0 induces a congruence relation on $T_\Sigma(V)$ as follows:

Definition 3.12. *For $s, t \in T_\Sigma(V)$, let $s \equiv_{T_0} t$ iff*

1. *$s, t \in T_{\Sigma_0}(V_0)$ and $T_0 \models s = t$ or*
2. *$s = t = x$ for an extended variable $x \in V \setminus V_0$ or*
3. *$s = f(s_1, \dots, s_k)$ and $t = f(t_1, \dots, t_k)$ for some operation symbol $f : w \to s$ in $\Sigma \setminus \Sigma_0$ and tuples $(s_1, \dots, s_k), (t_1, \dots, t_k) \in (T_\Sigma(V))^w$ such that $s_i \equiv_{T_0} t_i$ for all $i = 1, \dots, k$.*

T_0 can be enriched with the uninterpreted symbols of $\Sigma \setminus \Sigma_0$ giving rise to a new theory $T_0^=$, such that $T_0^= \models s = t$ iff $s \equiv_{T_0} t$.

Lemma 3.13 ([AR93])**.** *$\equiv_{T_0}$ is a congruence relation on $T_\Sigma(V)$.*

Proof. This is easily proved by induction on the height of terms. For further details see [AR93]. ☑

Henkin's version of the proof of Gödel's completeness theorem extends T_0 to a maximal consistent theory and enlarges it by adding formulae of the form $\exists x P(x) \Rightarrow P(c)$, where c is a new constant symbol called a *witness* of $\exists x P(x)$. Following Henkin's approach, the subsequent assumptions are presumed:

Assumption 3.14 (on the built-in theory and signature)**.** *T_0 is a complete theory and for every formula of the form $\exists x P(x)$ there is a ground term t in $T_{\Sigma_0}(\varnothing)$ such that $T_0 \models \exists x P(x) \Rightarrow P(t)$.*

If T_0 satisfies the above assumption then it is called a Henkin theory.

Remark 3.15. First of all, note that a complete theory is maximal consistent and all its models are elementary equivalent. Considering only complete theories is not a strong restriction since, usually, basic theories are understood as theories of a specific model. In particular, the examples of basic theories considered in this work, $\mathcal{PAR}$ and $\mathcal{SA}$, are complete.

As was mentioned in example 3.4, the constants $\dots, -1, 0, 1, \dots$ constitute the ground terms of the built-in language which give the universe of the models of interest. The method of construction of witnessing constants for a first-order language is a well-known way to expand the built-in language in order to obtain the required expressiveness. For the models of interest of the usual built-in theories ($\mathcal{SA}$, $\mathcal{PAR}$, etc.), it is always possible to give a signature with a ground term for any element of the universe. For example, with explicit constants $\dots, -1, 0, 1, \dots$ or with functions 's' for successor, 'p' for predecessor and constant '0'. $\lhd$

The preceding lemma gives rise to the construction of a Σ-algebra over T_0.

Lemma 3.16 ([AR93])**.** *Under the above assumption, $T_\Sigma(V)/\equiv_{T_0}$ is a Σ-algebra over T_0.*

Proof. (Sketch) The carrier of $T_\Sigma(V)/\equiv_{T_0}$ consists of the S-sorted set of equivalence classes of terms in $T_\Sigma(V)$ with respect to the congruence relation $\equiv_{T_0}$. Functions and predicates are interpreted according to representatives of equivalence classes. Because of completeness of T_0 and by induction on the structure of formulae it can be proved that the basic model of $T_\Sigma(V)/\equiv_{T_0}$ is also a model of the basic theory. For further details see [AR93]. ☑

Remark 3.17. In order to obtain from $T_\Sigma(V)$ (and the congruence $\equiv_{T_0}$) a Σ-algebra over T_0, T_0 should be a complete theory. In effect, suppose that T_0 is the (one-sorted) basic theory of Σ_0 (the language which consists of a predicate symbol p and two constant symbols a, b) axiomatized by the sentence $p(a) \vee p(b)$. By definition of the interpretation of predicate expressions, $p(a)$ and $p(b)$ do not hold in $T_\Sigma(V)$ since $T_0 \not\models p(a)$ and $T_0 \not\models p(b)$, respectively. Therefore the single axiom of T_0 does not hold in $T_\Sigma(V)$. Contrarily, if T_0 is assumed to be complete, then either $T_0 \models p(a)$ or $T_0 \models p(b)$ (or both), which implies that the interpretation of predicate expressions in $T_\Sigma(V)$ satisfies $p(a) \vee p(b)$. $\lhd$

Now, implications with built-in premises are defined.

Definition 3.18. *Let (Σ, Σ_0, T_0) be a signature. A universal Horn clause of the form:*

$$\forall X(t_1 = t_1' \wedge \cdots \wedge t_k = t_k' \wedge P \implies t_0 = t_0'),$$

where, for $i = 0, \ldots, k$, t_i, t_i' are S-sorted extended terms of the same sort and P is a built-in predicate, is called a universal Horn clause with built-in predicate P over the theory T_0. The built-in predicate P is often called built-in condition. $t_0 = t_0'$ is called the conclusion and $t_1 = t_1' \wedge \cdots \wedge t_k = t_k'$ is called the standard condition of the clause.

Definition 3.19. *Let $\Phi \equiv \forall X(t_1 = t_1' \wedge \cdots \wedge t_k = t_k' \wedge P \implies t_0 = t_0')$ be a universal Horn clause with built-in predicate and let A be an algebra over the built-in theory T_0. An assignment $\alpha : X \to A$ satisfies Φ iff its homomorphic extension, denoted also by $\alpha : T_\Sigma(X) \to A$, satisfies $\alpha(t_0) = \alpha(t_0')$ whenever $\alpha(t_m) = \alpha(t_m')$ for $m = 1, \ldots, k$ and $\mathfrak{A}$ satisfies P with assignment α (denoted also by $\mathfrak{A}_\alpha \models P$). Φ is said to be valid in an algebra A if all assignments $\alpha : X \to A$ satisfy Φ. This is denoted by $\mathrm{A} \models \Phi$.*

Given a set H of universal Horn clauses with built-in predicates over the theory T_0. The class of all algebras over T_0 which validate all clauses in H is denoted by $Mod_{T_0}\ H$. $Mod_{T_0}\ H$ is called the universal Horn class of H over the built-in theory T_0. $Mod_{T_0}\ H$ is said to be axiomatized by H and the clauses in H are said to be the axioms of $Mod_{T_0}\ H$.

A class $\mathcal{K}$ of algebras over T_0 is said to be a universal Horn class over the built-in theory T_0 if $\mathcal{K} = Mod_{T_0}\ H$ for some set H of universal Horn clauses with built-in predicates over T_0.

A clause Φ is said to be a logical consequence of H if for every algebra $\mathrm{A} \in Mod_{T_0}H$, $\mathrm{A} \models \Phi$. This is denoted by $H \models_{T_0} \Phi$ (or simply by $H \models \Phi$ when there is no confusion).

As usual, $\mathrm{A} \models_{T_0} H$ means that $\mathrm{A} \in Mod_{T_0}H$.

Example 3.20. Consider the signature of arrays presented in example 3.4. The following set of universal Horn clauses with built-in predicates over the Presburger arithmetic ($\mathcal{PAR}$) defines a universal Horn class over $\mathcal{PAR}$.

$$\forall i, j, A, \mathcal{L}\ (i =_{\mathcal{PAR}} j \Longrightarrow \langle A, i, \mathcal{L}\rangle[j] = \mathcal{L}).$$
$$\forall i, j, A, \mathcal{L}\ (i < j \vee j < i \Longrightarrow \langle A, i, \mathcal{L}\rangle[j] = A[j]).$$
$$\forall i, j, A, \mathcal{L}, \mathcal{K}\ (i =_{\mathcal{PAR}} j \Longrightarrow \langle\langle A, i, \mathcal{K}\rangle, j, \mathcal{L}\rangle = \langle A, j, \mathcal{L}\rangle).$$
$$\forall i, j, A, \mathcal{L}, \mathcal{K}\ (\neg(i =_{\mathcal{PAR}} j) \Longrightarrow \langle\langle A, i, \mathcal{K}\rangle, j, \mathcal{L}\rangle = \langle\langle A, j, \mathcal{L}\rangle, i, \mathcal{K}\rangle).$$
$$\forall i, j, k, A, \mathcal{L}, \mathcal{K}\ (\neg(i =_{\mathcal{PAR}} j) \Longrightarrow \langle\langle A, i, \mathcal{K}\rangle, k, \mathcal{L}\rangle[j] = \langle A, k, \mathcal{L}\rangle[j]).$$

i, j, k are (built-in) variables of sort *int*, A and $\mathcal{L}, \mathcal{K}$ of sort *array* and *element*, respectively. The formula $i =_{\mathcal{PAR}} j$ is an abbreviation for $i < j+1\ \&\ j < i+1$. Conditions of the fourth and fifth rules illustrate inequalities in the built-in language (this is not really necessary because $\mathcal{PAR} \models i \neq j \Leftrightarrow (i < j \vee j < i)$).

Note that universal Horn clauses as well as universal Horn clauses with built-in predicates admit only positive equalities as standard conditions. The intended semantics of $\langle A, i, \mathcal{L}\rangle$ is the array A replacing its i-th element with $\mathcal{L}$. The intended semantics of $A[i]$ is the i-th element of the array A. $\triangleleft$

Remark 3.21. The previous example appears to be non satisfactory because it does not combine built-in predicates and equations (as conditions). A more elaborated example, extension of the present one, combining equations and built-in predicates, in order to extend an order on the objects of the array sort to an order on the arrays is presented in [AR93]. So, it can be observed that in this sense the present approach is effectively superior to the use of constraints. $\triangleleft$

It remains to examine closure properties and algebraic characterizations of universal Horn classes over built-in theories, as is done for varieties and quasivarieties. As previously mentioned, subalgebras can be defined restricting the basic model of every subalgebra to remain identical to the original Σ-algebra. For this notion of subalgebra, one can reply the classical proof of closure under subalgebras for varieties and quasivarieties. Universal Horn classes over built-in theories are a generalization of universal Horn classes because universal Horn classes over the empty theory are precisely the second ones.

4. Calculus for universal Horn clauses with built-in predicates

A set of inference rules for universal Horn clauses with built-in predicates is introduced and its completeness is proved. Freeness notions for universal Horn clauses with built-in predicates are presented.

The following assumption is given to make decidability of built-in predicates effective.

Assumption 4.1 (on the basic theories)**.** *Only basic theories whose* set of quantifier-free logical consequences, $\boldsymbol{Th^{\forall}(T_0)} \stackrel{def}{=} \{P \mid T_0 \models P,\ P \ \textit{ is quantifier-free}\}$, *is decidable will be considered.*

Definition 4.2. *Let H be a set of universal Horn clauses with built-in predicates over a theory T_0. Consider the set of inference rules of Table 1. Deductions are finite sequences of universal Horn clauses with built-in predicates called formal proofs leading from the given set H to another clause Φ, the conclusion of the deduction. Clauses occurring in a formal proof are either in H or else can be inferred from earlier ones of the sequence by one of the inference rules. Φ is said to be deducible from H if there is a deduction from H whose last clause*

Rules of congruence:

$$\frac{}{\forall X(P)}, \qquad P \in Th^{\forall}(T_0),\ X \supseteq var(P).$$

$$\frac{}{\forall X(s{=}t)}, \qquad s \equiv_{T_0} t,\ X \supseteq var(s) \cup var(t).$$

$$\frac{}{\forall X(t{=}t' \implies t'{=}t)}.$$

$$\frac{}{\forall X(t{=}t' \wedge t'{=}t'' \implies t{=}t'')}.$$

$$\frac{}{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \implies f(\ldots,t_1,\ldots,t_n,\ldots){=}f(\ldots,t'_1,\ldots,t'_n,\ldots))}, \qquad \forall f \text{ in } \Sigma \smallsetminus \Sigma_0.$$

Rules of closure:

$$\frac{}{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \wedge P \implies t_1{=}t'_1)}, \qquad n \geq 1.$$

$$\frac{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \wedge P_1 \implies t{=}t')}{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_{n+k}{=}t'_{n+k} \wedge (P_1 \& P_2) \implies t{=}t')}, \qquad k, n \geq 0,\ (P_2 \text{ empty implies } k > 0).$$

$$\frac{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \wedge P_1 \implies t_0{=}t'_0), \forall X(t_0{=}t'_0 \wedge t_{n+1}{=}t'_{n+1} \wedge \cdots \wedge t_{n+k}{=}t'_{n+k} \wedge P_2 \implies t{=}t')}{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_{n+k}{=}t'_{n+k} \wedge (P_1 \& P_2) \implies t{=}t')},$$
$$k, n \geq 0.$$

$$\frac{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \wedge P_2 \implies t{=}t')}{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \wedge P_1 \implies t{=}t')}, \qquad P_1 \Rightarrow P_2 \in Th^{\forall}(T_0),\ n \geq 0.$$

$$\frac{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \wedge P \implies t{=}t')}{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \implies t{=}t')}, \qquad P \in Th^{\forall}(T_0),\ n \geq 0.$$

$$\frac{}{\forall X((\neg P) \implies t{=}t')}, \qquad P \in Th^{\forall}(T_0),\ X \supseteq var(P) \cup var(t) \cup var(t').$$

Rule of full invariance:

$$\frac{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \wedge P \implies t{=}t')}{\forall Y(\sigma(t_1){=}\sigma(t'_1) \wedge \cdots \wedge \sigma(t_n){=}\sigma(t'_n) \wedge \sigma(P) \implies \sigma(t){=}\sigma(t'))}, \qquad \forall \sigma : X \to T_\Sigma(V),$$
$$Y = \bigcup_{x \in X} var(\sigma(x)),\ n \geq 0.$$

Rules of conjunction:

$$\frac{\forall X(t_1{=}t'_1 \wedge t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \wedge P \implies t{=}t')}{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \wedge P \implies t{=}t')}, \qquad n \geq 1.$$

$$\frac{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \wedge P \implies t{=}t')}{\forall X(t_{\phi(1)}{=}t'_{\phi(1)} \wedge \cdots \wedge t_{\phi(n)}{=}t'_{\phi(n)} \wedge P \implies t{=}t')}, \ \forall \phi \text{ permutation of } \{1, \ldots, n\}\ (n \geq 2).$$

Rule of abstraction:

$$\frac{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \wedge P \implies t{=}t')}{\forall Y(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \wedge P \implies t{=}t')}, \quad \text{where } n \geq 0 \text{ and } Y = X \cup \{y\} \text{ for } y \in V \smallsetminus X.$$

Rule of concretion:

$$\frac{\forall X(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \wedge P \implies t_0{=}t'_0)}{\forall Y(t_1{=}t'_1 \wedge \cdots \wedge t_n{=}t'_n \wedge P \implies t_0{=}t'_0)}, \qquad \text{where } n \geq 0 \text{ and } Y = X \smallsetminus \{x\},$$
$$\text{for } x \in X \smallsetminus (\textstyle\bigcup_{i=0}^{n} var(t_i = t'_i) \cup var(P)),$$
$$X_s \neq \varnothing \implies T_\Sigma(Y)_s \neq \varnothing,\ \forall s \in S.$$

TABLE 1. Inference rules for universal Horn classes with built-in predicates.

is Φ. This is denoted by $H \vdash_{T_0} \Phi$ (or when no confusion can arise simply by $H \vdash \Phi$).

Remark 4.3. The first rule of congruence allows generalization of quantifier-free logical consequences of the built-in theory, i.e. valid built-in predicates. Note that formulae of the form $\forall X(P)$ are not universal Horn clauses with built-in predicates. This rule makes sense due to the assumption of decidability of $Th^{\forall}(T_0)$.

Note that the second rule of congruence trivially implies $\forall X(t = t)$. The fifth rule of congruence should be more precisely interpreted as

$$\forall X(t_1 = t'_1 \wedge \cdots \wedge t_n = t'_n \implies$$
$$f(s^0_1, \ldots, s^0_{k_0}, t_1, \ldots, t_n, s^n_1, \ldots, s^n_{k_n}) = f(s^0_1, \ldots, s^0_{k_0}, t'_1, \ldots, t'_n, s^n_1, \ldots, s^n_{k_n})),$$

where the terms t_i, t'_i for $1 \leq i \leq n$ are extended terms and the terms s^i_j for $0 \leq i \leq n,\ 1 \leq j \leq k_i$ are basic ones. By abuse of notation, if $k_i = 0$ for some $i = 0, \ldots, n$ then the corresponding sequence of basic arguments, $s^i_1, \ldots, s^i_{k_i}$, is considered empty.

The sixth rule of closure allows deduction of universal Horn clauses with T_0-inconsistent built-in predicates. In fact, P T_0-inconsistent means that any model of T_0 does satisfy P. Therefore, for all models $\mathfrak{A}$ of T_0, $\mathfrak{A} \models \neg P$ which means that $T_0 \models \neg P$. $\lhd$

There exists a smallest congruence relation on $T_\Sigma(V)$ generated by the set of universal Horn clauses H which includes $\equiv_{T_0}$. This congruence will be used to prove the completeness theorem.

Definition 4.4. *Let:*

- $\equiv^0_{H,T_0} = \equiv_{T_0}$
- $\equiv^{i+1}_{H,T_0}$ *is defined as the smallest congruence on $T_\Sigma(V)$ which includes $\equiv^i_{H,T_0}$ and such that for all clauses $\forall Y(t_1 = t'_1 \wedge \cdots \wedge t_n = t'_n \wedge P \implies t = t')$ in H and all assignments $\alpha : Y \to T_\Sigma(V)$ if $\alpha(t_i) \equiv^i_{H,T_0} \alpha(t'_i)$ for all $i = 1, \ldots, n$ and $T_0 \models \alpha(P)$ then $\alpha(t) \equiv^{i+1}_{H,T_0} \alpha(t')$.*

Finally, $\equiv_{H,T_0}$ is defined as $\bigcup_{i \geq 0} \equiv^i_{H,T_0}$.

Observe that $\equiv_{H,T_0}$ can also be conceived as the smallest congruence over the Σ-algebra $T_\Sigma(V)/\equiv_{T_0}$ generated by H, since the $\equiv^i_{H,T_0}$ congruences, for $i \geq 1$, do not change the congruence classes of basic sorts. Therefore it makes sense to speak of the congruence $\equiv_H$ over $T_\Sigma(V)/\equiv_{T_0}$. Obviously, $\equiv_{H,T_0}$ is the smallest congruence on $T_\Sigma(V)$ generated by H which contains $\equiv_{T_0}$. In effect, by definition $\equiv_{T_0}$ is contained in $\equiv_{H,T_0}$ and for all clauses $\forall Y(t_1 = t'_1 \wedge \cdots \wedge t_n = t'_n \wedge P \implies t = t')$ in H and all assignments $\alpha : Y \to T_\Sigma(V)$

if $\alpha(t_i) \equiv_{H,T_0} \alpha(t'_i)$ for all $i = 1, \ldots, n$ and $T_0 \models \alpha(P)$ then exists k such that $\alpha(t_i) \equiv^k_{H,T_0} \alpha(t'_i)$ for all $i = 1, \ldots, n$ and by definition $\alpha(t) \equiv^{k+1}_{H,T_0} \alpha(t')$ which implies that $\alpha(t) \equiv_{H,T_0} \alpha(t')$.

A freeness notion for universal Horn classes with built-in predicates is examined before the completeness proof.

Results about freeness and initiality of $T_\Sigma(V)/\equiv_H$ and $T_\Sigma/\equiv_H$ for the multi-sorted case (when T_0 is the empty theory) can be found in [Kap83] and [Cla88]. These results cannot directly be applied for the case of universal Horn clauses with built-in predicates because $T_\Sigma(V)/\equiv_{H,T_0}$ carries the semantics of the built-in theory T_0.

First of all, recall that only complete theories are considered. Usually one works with a special model of the theory, for example for the case of the $\mathcal{ANT}$ only the *standard model* and its isomorphisms are considered, i.e., the model over the structure of the natural numbers with the usual addition and successor.

In the following two theorems $Mod_{T_0}H$ is restricted to the subclass of Σ-algebras whose reduct with respect to the basic part (Σ_0) is a term generated model of T_0 (see [Gal87] for definitions). This subclass of Σ-algebras is called basic-term generated algebras of $Mod_{T_0}H$. In a basic-term generated algebra A of $Mod_{T_0}H$, every function and every predicate symbol in Σ_0 receive an interpretation in $\mathfrak{A}$. In particular, the standard models of $\mathcal{SA}$ and $\mathcal{ANT}$ are term generated (in effect, the interpretations $0^{\mathfrak{A}}, s(0)^{\mathfrak{A}}, s(s(0))^{\mathfrak{A}}, \ldots$ represent all possible elements of an standard model $\mathfrak{A}$ of $\mathcal{SA}$ and $\mathcal{ANT}$).

Theorem 4.5 (Construction of free algebras, [AR93]). *$T_\Sigma(V)/\equiv_{H,T_0}$ is free over V in the subclass of basic-term generated algebras of $Mod_{T_0}H$.*

Proof. (Sketch) To see that $T_\Sigma(V)/\equiv_{H,T_0} \in Mod_{T_0}H$, observe that lemma 3.16 and the fact that $\equiv_{H,T_0}$ does not change the congruence classes of basic terms of $\equiv_{T_0}$ imply that $T_\Sigma(V)/\equiv_{H,T_0}$ is a Σ-algebra over T_0. Let $\forall X(t_1 = t'_1 \wedge \cdots \wedge t_n = t'_n \wedge P \implies t = t')$ be a clause in H and $\alpha : X \to T_\Sigma(V)/\equiv_{H,T_0}$ be an arbitrary assignment and suppose that its homomorphic extension, de noted also by $\alpha : T_\Sigma(X) \to T_\Sigma(V)/\equiv_{H,T_0}$, satisfies $\alpha(t_i) = \alpha(t'_i)$ for all $i = 1, \ldots, n$ and $T_0 \models \alpha(P)$. Let $nat : T_\Sigma(V) \to T_\Sigma(V)/\equiv_{H,T_0}$ be the natural application from terms into its corresponding equivalence classes with respect to $\equiv_{H,T_0}$. Since nat is surjective there exists $h : X \to T_\Sigma(V)$ such that its homomorphic extension, denoted also by $h : T_\Sigma(X) \to T_\Sigma(V)$, satisfies $\alpha = itnat \circ h$. Then $nat(h(t_i)) = nat(h(t'_i))$ for all $i = 1, \ldots, n$. Therefore $h(t_i) \equiv_{H,T_0} h(t'_i)$. By construction of $\equiv_{H,T_0}$ and since $T_0 \models h(P)$, there exists k such that $h(t_i) \equiv^k_{H,T_0} h(t'_i)$ for all $i = 1, \ldots, n$. Consequently $h(t) \equiv^{k+1}_{H,T_0} h(t')$ which implies that $\alpha(t) = nat(h(t)) = nat(h(t')) = \alpha(t')$.

To prove that for all basic-term generated algebras A of $Mod_{T_0}H$ and arbitrary assignments $\alpha : V \to A$ there exists a unique $\beta : T_\Sigma(V)/\equiv_{H,T_0} \to A$ such that $\beta \circ nat = \alpha$ steps of freeness p roof in [EM85] could be followed making appropriate considerations on $\equiv_{T_0,H}$. For further details see [AR93]. ☑

Theorem 4.6 ([AR93]). *Let $\forall X(t = t')$ be a purely equational clause which holds in the Σ-algebra $T_\Sigma(X)/\equiv_{H,T_0}$. Then for all basic-term generated algebras* A *of $Mod_{T_0}H$,* A $\models \forall X(t = t')$.

Proof. Let $\alpha : X \to A$ be an arbitrary assignment and $\alpha : T_\Sigma(X) \to A$ denote also its homomorphic extension. By the previous theorem, there exists exactly a homomorphism $\beta : T_\Sigma(X)/\equiv_{H,T_0} \to A$, such that $\alpha = \ beta \circ nat$, where $nat : T_\Sigma(X) \to T_\Sigma(X)/\equiv_{H,T_0}$ is the natural application from terms into its corresponding equivalence classes with respect to $\equiv_{H,T_0}$.

Validity of $\forall X(t = t')$ in $T_\Sigma(X)/\equiv_{H,T_0}$ implies $nat(t) = nat(t')$ and then $\beta(nat(t)) = \beta(nat(t'))$. Thus $\alpha(t) = \alpha(t')$. Therefore A $\models \forall X(t = t')$. ☑

Remark 4.7. In the preceding two theorems the essential restriction of $Mod_{T_0}H$ to basic-term algebras allows a treatment of Σ-homomorphisms similar to the classical one in the algebraic context. In a certain sense, this restriction can be thought as a type of initial property for the basic theory, since only Σ-algebras whose basic part is a standard model generated by the language of Σ_0 are admitted. To illustrate precisely the above situation, consider $Mod_{\mathcal{SA}}H$ (without the mentioned restriction), where the basic theory is $\mathcal{SA}$, the extension $\Sigma \setminus \Sigma_0$ is composed by two constant symbols $a, b : \to s$ and a function symbol $f : nat \to s$, where nat is the sort of $\mathcal{SA}$ and s is a new sort and H is the following set of clauses:

$$f(0) = a;$$
$$\forall x(f(x) = a \Rightarrow f(s(x)) = a).$$

Note that the equation $\forall x(f(x) = a)$ does not hold in $Mod_{\mathcal{SA}}H$ since there exists a nonstandard model (of $\mathcal{SA}$) which satisfies $f(x) = a$ for all $x \in I\!N$ and $f(x) = b$ for all $x \notin I\!N$. Therefore $H \not\models \forall x(f(x) = a)$. However, note that $T_\Sigma(V)/\equiv_{H,T_0} \models \forall x(f(x) = a)$. On the other hand, $H \not\vdash \forall x(f(x) = a)$, since the induction schema of $\mathcal{SA}$ does not apply for the whole extended language (viz, $a, b, f \in \Sigma \setminus \Sigma_0$).

In the case of $\mathcal{PAR}$ the situation is more complex because induction is equivalent to the well-ordering property for integers, namely, that every nonempty set of integers with a lower bound has a least element. ◁

The completeness proof of the inference rules is resumed with The following simple lemma.

Lemma 4.8. *Let H be a set of universal Horn clauses. Let C and P be respectively any fixed standard condition and built-in predicate. Let $\sim_{C,P}$ be defined as $t \sim_{C,P} t'$ iff $H \vdash \forall X(C \wedge P \implies t = t')$. Then $\sim_{C,P}$ is an equivalence relation.*

Theorem 4.9 (Completeness). *Let H be a set of universal Horn clauses with built-in predicates over a theory T_0. An arbitrary universal Horn clause Φ with built-in predicates over T_0 is a logical consequence of H if and only if Φ is deducible from H. In symbols:*

$$H \models \Phi \quad \textit{iff} \quad H \vdash \Phi.$$

Proof. (Sketch) Soundness of the calculus can easily be showed for each rule, as usual.

Conversely, to prove that $H \models \Phi \implies H \vdash \Phi$, let $C = s_1 = s_1' \wedge \cdots \wedge s_k = s_k'$ be any standard condition with terms in $T_\Sigma(X)$ and let P be any T_0-consistent built-in predicate with terms in $T_{\Sigma_0}(X_0)$, where X_0 denotes the basic variables of X. Note that by the sixth rule of closure it is enough to consider T_0-consistent built-in predicates. $sk(t)$ denotes the term t considering the variables of X as constants. Let $\mathrm{A}_{X,C\wedge P}$ be the quotient algebra $T_\Sigma(X)/\equiv_{\tilde{H},\tilde{T_0}}$, where $\tilde{H} = H \cup sk(C)$ and $\tilde{T_0} = T_0 \cup \{sk(P)\}$. Note that to speak of $T_\Sigma(X)$ with the semantics of $T_0 \cup \{sk(P)\}$ makes sense, because for a quantifier-free formula Φ in $T_{\Sigma_0}(X_0)$ one can decide if $T_0 \models sk(P) \Rightarrow \Phi$ and this is equivalent to $T_0 \cup \{sk(P)\} \models \Phi$ by the well known deduction theorem. The definition of $\equiv_{\tilde{H},\tilde{T_0}}$ is analogous with the one of $\equiv_{H,T_0}$.

It is enough to prove that $\mathrm{A}_{X,C\wedge P} \models sk(t) = sk(t')$ implies $H \vdash \forall X(C \wedge P \implies t = t')$. In effect, assume that this holds and suppose that for every algebra $\mathrm{A} \in Mod_{T_0}H$, $\mathrm{A} \models \forall X(C \wedge P \implies t = t')$. Then in particular $\mathrm{A}_{X,C\wedge P} \models \forall X(C \wedge P \implies t = t')$ and, by definition of $\mathrm{A}_{X,C\wedge P}$, $\mathrm{A}_{X,C\wedge P} \models sk(C) \wedge sk(P)$ which implies $\mathrm{A}_{X,C\wedge P} \models sk(t) = sk(t')$. Then by assumption $H \vdash \forall X(C \wedge P \implies t = t')$. Note that $\mathrm{A}_{X,C\wedge P}$ is effectively an algebra in $Mod_{T_0}H$: by lemma 3.16 and since P is T_0-consistent, $T_\Sigma(V)/\equiv_{\tilde{T_0}}$ is a Σ-algebra over T_0; by definition of $\equiv_{H,T_0}$ and in particular of $\equiv_{\tilde{H},\tilde{T_0}}$, $\mathrm{A}_{X,C\wedge P}$ validates all the clauses in H.

A family of congruences similar to $\equiv^i_{H,T_0}$ is built in order to prove that $\mathrm{A}_{X,C\wedge P} \models sk(t) = sk(t')$ implies $H \vdash \forall X(C \wedge P \implies t = t')$.

Let $\equiv_0^{C\wedge P}$ be the smallest congruence on $T_\Sigma(X)$ (which in this case carries the semantics of $\tilde{T_0}$) which contains $\equiv_{\tilde{T_0}}$ and $sk(C)$ and such that for all purely equational clauses $\forall Y(u = u') \in H$ and all assignments $\alpha : Y \to T_\Sigma(X)$, $sk(\alpha(u)) \equiv_0^{C\wedge P} sk(\alpha(u'))$.

$\equiv_{i+1}^{C\wedge P}$ is recursively defined as the smallest congruence on $T_\Sigma(X)$ which includes $\equiv_i^{C\wedge P}$ and such that for all universal Horn clauses $\forall Y(t_1 = t'_1 \wedge \cdots \wedge t_n = t'_n \wedge Q \implies t = t')$ in H and all assignments $\alpha : Y \to T_\Sigma(X)$, if $T_0 \models sk(P) \Rightarrow sk(\alpha(Q))$ and $sk(\alpha(t_i)) \equiv_i^{C\wedge P} sk(\alpha(t'_i))$ for all $i = 1, \ldots, n$ then $sk(\alpha(t)) \equiv_{i+1}^{C\wedge P} sk(\alpha(t'))$.

Finally, $\equiv^{C\wedge P} = \bigcup_{i\geq 0} \equiv_i^{C\wedge P}$. To conclude it, it is sufficient to prove that for all $i \geq 0$, $sk(t) \equiv_i^{C\wedge P} sk(t')$ implies $H \vdash \forall X(C \wedge P \implies t = t')$. This results from a relatively simple inductive proof.

i=0: By the lemma 4.8 and fifth rule of congruence it is enough to prove that $H \vdash \forall X(C \wedge P \implies \alpha(t) = \alpha(t'))$ for all equations $\forall Y(t = t')$ in $H \cup sk(C)$ and all assignments $\alpha : Y \to T_\Sigma(X)$.

First, suppose that $\forall Y(t = t') \in H$. By the rule of full invariance $H \vdash \forall X(\alpha(t) = \alpha(t'))$ and by the second rule of closure $H \vdash \forall X(C \wedge P \implies \alpha(t) = \alpha(t'))$. Second, let $s_j = s'_j \in C$ (for some $1 \leq j \leq k$). By the first rule of closure $H \vdash \forall X(C \wedge P \implies s_j = s'_j)$.

i: Suppose that $sk(t) \equiv_i^{C\wedge P} sk(t')$ implies $H \vdash \forall X(C \wedge P \implies t = t')$.

i+1: Again, by lemma 4.8 and the fifth rule of congruence it is enough to prove that $H \vdash \forall X(C \wedge P \implies \alpha(t) = \alpha(t'))$ for all universal Horn clauses $\forall Y(t_1 = t'_1 \wedge \cdots \wedge t_n = t'_n \wedge Q \implies t = t')$ in H and all assignments $\alpha : Y \to T_\Sigma(X)$ such that $sk(\alpha(t_j)) \equiv_i^{C\wedge P} sk(\alpha(t'_j))$ holds for all $j = 1, \ldots, n$ and $T_0 \models sk(P) \Rightarrow sk(\alpha(Q))$.

By the rule of full invariance, $H \vdash \forall X(\alpha(t_1) = \alpha(t'_1) \wedge \cdots \wedge \alpha(t_n) = \alpha(t'_n) \wedge \alpha(Q) \implies \alpha(t) = \alpha(t'))$. By induction hypothesis $H \vdash \forall X(C \wedge P \implies \alpha(t_j) = \alpha(t'_j))$ for all $i = 1, \ldots, n$. Applying repeatedly the third rule of closure and the rules of conjunction, it can be obtained $H \vdash \forall X(C \wedge (P \& \alpha(Q)) \implies \alpha(t) = \alpha(t'))$. Note also that $T_0 \models P \Rightarrow \alpha(Q)$ results trivially as a generalization of $T_0 \models sk(P) \Rightarrow sk(\alpha(Q))$ because in a given model $\mathfrak{A}$ of T_0 with universe A any interpretation of the constant symbols of $sk(P)$ (i.e. of the basic variables of X considered as constants) is permissible. Finally, by applying the fourth rule of closure to $H \vdash \forall X(C \wedge (P \& \alpha(Q)) \implies \alpha(t) = \alpha(t'))$ and the fact that $T_0 \models P \Rightarrow (P \& \alpha(Q))$ (which results trivially from $T_0 \models P \Rightarrow \alpha(Q)$) it is obtained $H \vdash \forall X(C \wedge P \implies t = t')$. ☑

5. **Conclusions**

Conditional equational specifications and built-in predicates or constraints in a many-sorted first-order theory were carefully combined. This amalgamation of algebraic structures and logical theories is useful in the fields of algebraic specification and term rewriting but it is usually treated without giving enough attention to the related formal problems. In fact, some algebraic properties such as freeness and the simple notion of initial algebra cannot naturally be inherited from purely conditional equational specifications, as it is frequently assumed. Here special and appropriate notions and restrictions on the built-in predicates and on the conditional equational specifications were developed guaranteeing the basic algebraic properties needed in order to obtain a complete deductive calculus for this class of combined specifications.

In order to make the process of deduction effective, standard conditional rewriting techniques and built-in decision algorithms should appropriately be combined. Appropriate notions of standard conditional term rewriting systems with built-in predicates appear in [AR97] where it was reported on a *complete* algorithm based on conditional term rewriting and built-in decision algorithms for effective deduction of formulae of the form $P \Longrightarrow s = t$, where P is a pure built-in premise. The author presents also proofs of Newman's diamond lemma and the Church-Rosser property for the corresponding class of combined term rewriting systems in [AR98].

Acknowledgements. The author would like to thank Claus-Peter Wirth and Klaus Madlener for their valuable suggestions and comments which have improved his early conception of this work and Rodolfo Baigorri for his corrections and remarks.

References

[AB92] J. Avenhaus and K. Becker, *Conditional rewriting modulo a built-in algebra*, SEKI-Report SR-92-11, Fachbereich Informatik, Universität Kaiserslautern, Postfach 3049, D-67653 Kaiserslautern (Germany), 1992.

[AR93] M. Ayala-Rincón, *Expressiveness of Conditional Equational Systems with Built-in Predicates*, Ph.D. thesis, Universität Kaiserslautern, Kaiserslautern (Germany), December 1993.

[AR94] M. Ayala-Rincón, *Confluence of Conditional Rewriting Systems with Built-in Predicates and Standard Premises as Conditions*, XXI Seminário Integrado de Software e Hardware, Caxambú, Brazil, August 1994, pp. 507–521.

[AR95] M. Ayala-Rincón, *A Deductive Calculus for Conditional Equational Systems with Built-in Predicates as Premises — Extended Abstract —*, XV International Conference of the Chilean Computer Science Society, Arica, Chile, November 1995, pp. 25–36.

[AR97] M. Ayala-Rincón, *A Decision Procedure for Conditional Rewriting Systems with Built-in Predicates*, XXIV Seminário Integrado de Software e Hardware, Brasília, Brazil, August 1997, pp. 387–398.

[AR98] M. Ayala-Rincón, *Church-Rosser Property for Conditional Rewriting Systems with Built-in Predicates as Premises*, Second Frontiers of Combining Systems, Amsterdam, Holland, Applied Logic Series, Kluwer Academic Publishers, October 1998, To appear.

[ARG97] M. Ayala-Rincón and L. M. R. Gadelha, *Some Applications of (Semi-)Decision Algorithms for Presburger Arithmetic in Automated Deduction based on Rewriting Techniques*, La Revista de La Sociedad Chilena de Ciencia de la Computación **2** (1997), no. 1, 14–23.

[Bec94] K. Becker, *Rewrite Operationalization of Clausal Specifications with Predefined Structures*, Ph.D. thesis, Universität Kaiserslautern, Kaiserslautern (Germany), April 1994.

[CK90] C. C. Chang and H. J. Keisler, *Model Theory*, third ed., Studies in Logic and the Foundation of Mathematics, vol. 73, North-Holland Publishing Company, 1990.

[Cla88] I. Classen, *Algebraische Grundlagen der Termersetzung mit bedingten Gleichungen*, Tech. Report 88-04, TU Berlin, 1988, In German.

[Coo72] D. C. Cooper, *Theorem Proving in Arithmetic without Multiplication*, Machine Intelligence **7** (1972), 91–99.

[DO90] N. Dershowitz and M. Okada, *A Rationale for Conditional Equational Programming*, Theoretical Computer Science **75** (1990), 111–138.

[EM85] H. Ehrig and B. Mahr, *Fundamentals of Algebraic Specification 1*, EATCS Monographs on Theoretical Computer Science, Springer, 1985.

[Gal87] J. H. Gallier, *Logic for Computer Science: Foundations of Automatic Theorem Proving*, John Wiley & Sons, Inc., 1987.

[Kap83] S. Kaplan, *Un langage de Spécification de types abstraits algébriques*, Ph.D. thesis, Universite de Paris-Sud Centre D'Orsay, February 1983, In French.

[Kap84] S. Kaplan, *Conditional Rewrite Rules*, Theoretical Computer Science **33** (1984), 175–193.

[KKR90] Claude Kirchner, Hélène Kirchner, and M. Rusinowitch, *Deduction with Symbolic Constraints*, Revue d'Intelligence Artificielle **4** (1990), no. 3, 9–52, Special issue on Automatic Deduction.

[O'D77] M. J. O'Donnell, *Computing in Systems Described by Equations*, LNCS, vol. 58, Springer, 1977.

[Pad88] P. Padawitz, *Computing in horn clause theories*, EATCS Monographs on Theoretical Computer Science, Springer, 1988.

[Pre29] M. Presburger, *Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt*, 1. Kongres matematyków krajow slowiańskich, Warsaw, 1929, in German, pp. 92–101.

[Sho81] R. E. Shostak, *Deciding Linear Inequalities by Computing Loop Residues*, Journal of the Association for Computing Machinery **28** (1981), no. 4, 769–779.

[Vor89] S. G. Vorobyov, *Conditional rewrite rule systems with Built-in Arithmetic and Induction*, Proc. Third Int. Conference on Rewriting techniques and Applications, Chapel-Hill, NC (N. Dershowitz, ed.), LNCS, vol. 355, Springer, April 1989, pp. 492–512.

[Wec92] W. Wechler, *Universal algebra for computer scientists*, EATCS Monographs on Theoretical Computer Science, Springer, 1992.

(Recibido en abril de 1996, revisado en noviembre de 1997)

Departamento de Matemática
Universidade de Brasília
70900-010 Brasília DF, BRASIL
e-mail: ayala@mat.unb.br