Abstract Computability, Algebraic Specification and Initiality

J.V. Tucker

Department of Computer Science,
University of Wales, Swansea SA2 8PP, Wales
J.V.Tucker@swansea.ac.uk

J.I. Zucker*

Department of Computing and Software,
McMaster University, Hamilton, Ont. L8S 4L7, Canada
zucker@mcmaster.ca

Abstract

Abstract computable functions are defined by abstract finite deterministic algorithms on many-sorted algebras. We show that there exist finite universal algebraic specifications that specify uniquely (up to isomorphism) (*i*) all absract computable functions on any many-sorted algebra; and (*ii*) all functions effectively approximable by abstract computable functions on any metric algebra. We show that there exist universal algebraic specifications for all the classically computable functions on the set $\mathbb{R}$ of real numbers. The algebraic specifications used are mainly bounded universal equations and conditional equations. We investigate the initial algebra semantics of these specifications, and derive situations where algebraic specifications define precisely the computable functions.

Categories and Subject Descriptors: F.1.1 (**Computation by Abstract Devices**): Models of Computation — *computability theory*; F.4.1 (**Mathematical Logic and Formal Languages**): Mathematical Logic — *computability theory; proof theory*

General Terms: Abstract Computability, Algebraic Specification, Computable Analysis, Conditional Equations, Equational Logic, Metric Algebras

Additional Key Words and Phrases: Birkhoff's theorem, Grzegorczyk-Lacombe computability, initial algebras, term models, many-sorted algebras, Mal'cev's theorem, topological algebras

0 Introduction

Abstract computability theory is the theory of computable functions and relations over many-sorted algebras. It is a generalisation of classical recursion theory on the natural numbers, based on notions of finite deterministic computation on an arbitrary many-sorted algebra. An important feature of the theory is its analysis of computations that are *uniform* over classes of algebras, and a natural application of the theory is to analyse the scope and limits of models of computation and specification over abstract data types and their implementations. Since the 1960s, many abstract models of computation have been defined and classified, starting with the models of E. Engeler, Y. Moschovakis, H. Friedman and J.C. Shepherdson, and generalised Church-Turing Theses for computation and specification have been formulated and defended [TZ88, TZ92]. Here we will use the model of computation μPR^* (a generalised form of Kleene schemes), which involves *simultaneous primitive recursion* and *least number search* over a many-sorted algebra augmented by the booleans, natural numbers and finite sequences of every sort. In [TZ88] the model μPR^* is shown to be equivalent to 'while'-array programs over these algebras, the primary mathematical model of imperative programming.

Working with finite computation on any algebra enables us to develop a number of special computability theories for algebras, such as *rings and fields of real numbers* [Tuc80, Eng93, BSS89, BCSS96, BCSS97] and *topological and metric algebras* [TZ99]. For a comprehensive introduction to abstract computability, including a survey of its origins in the 1950s and principal literature, see our survey [TZ00].

In this paper we prove theorems that show that functions that are abstractly computable over many-sorted algebras, or have abstractly computable *approximations* on topological algebras, can be specified by purely algebraic methods, but that the converse does not hold in the absence of certain topological conditions.

Algebraic specification methods characterise functions as the solutions of systems of algebraic formulae; normally, the solutions are unique. By algebraic formulae, we mean equations

$$t(\mathbf{x}) \;=\; t'(\mathbf{x})$$

or conditional equations

$$t_1(\mathbf{x}) = t'_1(\mathbf{x}) \wedge \ldots \wedge t_k(\mathbf{x}) = t'_k(\mathbf{x}) \quad \longrightarrow \quad t(\mathbf{x}) = t'(\mathbf{x}), \qquad\qquad (*)$$

or, more generally, conditional formulae

$$R_1 \wedge \ldots \wedge R_k \quad \longrightarrow \quad R \qquad\qquad (**)$$

where the formulae R_i and R are generalisations of equations, making use of the distinguished sorts nat of naturals and real of reals (as we will see below). To define a unique solution for a system of equations, in logic one often thinks of *definability up to isomorphism*, and in computing one often thinks in terms of *initial algebra semantics* (or possibly *final algebra semantics*). However, notice that there are many more equational methods, *e.g.*, for specifying concurrent processes using metric space methods to solve equations [dBR92, dBdV99], or for computing solutions of differential or integral equations.

In computation over a many-sorted algebra A we use the booleans, natural numbers and finite sequences over A. With regard to algebraic specifications over such structures, generalising conditional equations leads to the concept of *conditional bounded universal (BU) equations*, in which the formulae R_i and R of $(**)$ may have the form

$$t_1 = t_2 \qquad \text{or} \qquad \forall \mathsf{z} < t\,[\,t_1 = t_2\,]$$

where the variable z and term t are of sort nat.

Conditional BU equations are new and provide us with more appropriate axiomatisations for some properties using the natural number sort; we show they are equivalent with conditional equations. The main theorems are first proved for conditional BU equations and the reduction method applied to obtain conditional equational specifications.

In the first part of the paper, we begin with the "simple" situation where there is a system E of conditional equations over a signature Σ, and a Σ-algebra A such that E has one and only one solution f on A. We call this method of characterising functions *conditional equation definability* on A. We address the obvious general question:

Does abstract computability imply conditional equation definability?

The answer is yes, and we show that there exist *universal specifications* that specify *all* computable functions, as follows (Section 5, Theorem 4).

Theorem A (Algebra). *Given a signature Σ and function type τ over Σ, there exists a finite set of conditional equations $E(\mathsf{z})$ (with a distinguished natural number variable z) over a finite expansion Σ' of Σ, such that for any abstract program α over Σ, if A is any Σ-algebra and f a total function on A of type τ computed by α, then f is defined uniquely on A by $E(\bar{k})$, where $\bar{k}$ is a numeral instantiating z which is effectively calculable from α. The system $E(\mathsf{z})$ is uniformly computable from Σ and τ.*

Applying our abstract computability theory to *metric algebras*, we can obtain an important, strictly broader, class of functions: namely, those *uniformly approximable by abstractly computable functions*. In metric algebras, approximation is elegantly formulated in terms of the distance function, which uses the sort real. This gives rise to a broader class of conditional formulae than $(*)$, called *conditional equations and inequalities*, namely formulae $(**)$ in which the formulae R_i and R may have the form

$$t_1 = t_2 \qquad \text{or} \qquad t_1 < t_2$$

where, in the case of inequality $(t_1 < t_2)$, t_1 and t_2 are of sort real.

From Theorem A we then prove (Section 6, Theorem 2):

Theorem B (Metric algebra). *Given a signature Σ and function type τ over Σ, there exists a finite set of conditional equations and inequalities $E(\mathsf{z})$ (with a distinguished natural number variable z) over a finite expansion Σ' of Σ, such that for any abstract program α over Σ, if A is any metric Σ-algebra and f a total function on A of type τ, approximable by α in the following sense: for all $a \in A$ and all n*

$$d(f(a), [\![\alpha]\!](n, a)) < 2^{-n},$$

then f is defined uniquely on A by $E(\bar{k})$, where $\bar{k}$ is a numeral instantiating $\mathbf{z}$ which is effectively calculable from α. The system $E(\mathbf{z})$ is uniformly computable from Σ and τ.

Thus, *there is a bound $B(\Sigma, \tau)$ on the number of conditional equations and inequalities needed to define all computable or computably approximable functions, that depends only on the signature Σ and the function type τ.*

Using Theorem B, we show that all the classically computable functions of real analysis are unique solutions of finite sets of conditional equations and inequalities. These classically computable functions have several characterisations, starting with those of Grzegorczyk [Grz55, Grz57] and Lacombe [Lac55], and hence are often called *GL-computable*. The familiar functions of analysis, such as $\sin x$, e^x, $\log x$, etc., are all GL-computable. Now there exists a certain simple total metric algebra $\mathcal{I}^d$ over the real unit interval $I = [0, 1]$, such that the total functions on I which are uniformly approximable by abstractly computable functions on $\mathcal{I}^d$ are precisely the GL-computable functions on I [TZ99, TZ00]. We prove the following (Section 6, Theorem 3).

Theorem C (Metric algebra over a real interval). *For each positive integer m there is a signature Σ_m^* which is an expansion of the signature of $\mathcal{I}^d$ by finitely many function symbols, and a finite system of conditional equations and inequalities $E_m(\mathbf{z})$ (with a distinguished natural number variable $\mathbf{z}$) over Σ_m^*, such that any total function $f : [0, 1]^m \to \mathbb{R}$ that is GL-computable, is the unique solution of $E(\bar{k})$ for some substitution of a numeral $\bar{k}$ for $\mathbf{z}$. The specification $(\Sigma_m^*, E_m(\mathbf{z}))$ is uniformly computable from m.*

Thus *there is a bound $B(m)$ on the number of conditional equations and inequalities needed to define all m-ary GL-computable functions on $[0, 1]$.*

The signature Σ_m^* consists of the sorts of booleans $\mathbb{B}$ and naturals $\mathbb{N}$, with their standard operations; the sort of reals $\mathbb{R}$, with its ring operations, together with division of reals by naturals; the sort of the unit interval I, with its embedding into $\mathbb{R}$; the sort of finite arrays on $\mathbb{R}$ with their standard operations; the standard metrics on all these sorts; a "universal function" which approximably abstractly computes all m-ary GL-computable total functions on I, together with the auxiliary functions used in its computation; the function 2^{-n} used for expressing approximations; and a function for computing bounded quantification over $\mathbb{N}$.

This theorem has some interesting consequences, one of which we illustrate (Section 6, Theorem 4):

Corollary. *For each $n > 0$, there is a finite universal algebraic specification, consisting of conditional equations and inequalities, for all computable finite dimensional dynamical systems on the unit n-cube and over the unit time interval.*

Next we consider the converse problem:

Problem. *Find (reasonable) conditions under which algebraic definability implies abstract computability.*

From Theorem C it follows that the converse to Theorem A is false, at least for specifications consisting of conditional equations and inequalities; for example, for the sine and

cosine functions on the unit interval.

It is an open problem whether the converse of the approximation result (Theorems B and C) holds. It seems that some extra *topological condition* such as continuity is required for a converse result. This suggests an interesting research area; see the example and discussion in Section 6.3.

In the second part of the paper, we show how the conditional equational theories, and conditional BU equational theories, can be used with standard algebraic specification methods associated with proof systems, term rewriting and initial algebra semantics.

Now, when using the booleans, natural numbers and finite sequences, the algebraic specifications and their initial algebra semantics must define the corresponding standard models of the booleans, natural numbers and finite arrays. We develop extensions of the Birkhoff-Mal'cev Completeness Theorems that underlie the algebraic specification methods, designed to ensure that these sorts have standard models. Then we prove (Section 8, Theorems 4 and 5):

Theorem D (Initial algebra). *Given a signature Σ and function type τ over Σ, there exists a finite set of conditional equations $E(\mathsf{z})$ (with a distinguished natural number variable z) over a finite expansion Σ' of Σ, such that for any abstract program α over Σ, if α computes a total function f on A of type τ, and A has an initial algebra specification by a set E of either conditional equations or conditional BU equations (with hidden sorts and functions), then (A, f) has an initial algebra specification by a set $E \cup E(\bar{k})$, where $E(\mathsf{z})$ consists of conditional equations and $\bar{k}$ is a numeral instantiating z which is effectively calculable from α. The system $E(\mathsf{z})$ is uniformly computable from Σ and τ. Furthermore, if the specification E of A has e axioms, then the specification of (A, f) is finite, with $e + e'$ axioms, where e' is a constant computed uniformly from Σ and τ.*

This paper is part of our series on abstract computability theory on many-sorted algebras and its applications, starting in [TZ88] and most recently surveyed in [TZ00]. Knowledge of computation and our studies of computation versus specifications [TZ92, TZ91] and verification [TZ93] will be helpful, but only our work on topological data types [TZ99] is necessary.

The subject of this paper is also a generalisation of the theory of algebraic specifications for *computable, semicomputable and co-semicomputable* algebras developed by one of us (JVT) with J.A. Bergstra: see [BT80b, BT80a, BT82, BT83, BT87, BT95] and the surveys [MG85, SHT95]. However, at least initially, the generalised computability raises new questions concerning topological data types, uniformity and parameterisation, and standard models. Knowledge of the theory for computable algebras is not required for this paper.

In Section 1 we define how to augment structures with the *standard sorts* of the booleans and naturals, and finite sequences or *arrays* over all sorts, together with the corresponding operations. For the rest of the paper we consider, without loss of generality, only N-standard signatures and structures with the booleans and naturals.

In Section 2 we introduce a number of proof systems, all based in the calculus of se-

quents over a many-sorted signature Σ. These are systems for (i) first order logic over Σ with equality, (ii) conditional equational logic, (iii) conditional bounded universal (BU) equational logic, and (iv) conditional standard universal (SU) equational logic. The systems (ii) and (iii) are subsystems of the classical predicate calculus (i), and are used in the following sections, while (iv) is an infinitary system introduced for interest.

In Section 3 we define the basic technical notion of a theory uniquely specifying a function on an arbitrary algebra with hidden sorts and functions. This leads to a simple notion of specifiable parameterisation which we illustrate by showing how a conditional equational (or conditional BU equational) specification of a standard structure A can be extended to a similar specification of the array structure A^*. We also show how to "reduce" a conditional BU equational specification over Σ to a conditional equational specification over an expansion of Σ.

In Section 4 we recall the basic notions of *computability* of functions, including universality of the μPR^* functions.

In Section 5 we prove Theorem A above, concerning the conditional equational definability of computable functions.

In Section 6 we prove Theorems B and C, concerning the definability, by conditional equations and inequalities, of computably approximable functions on metric algebras.

In Section 7 we describe the construction of *initial standard models* for conditional equational and conditional BU equational theories, and work out the completeness theorems for the corresponding proof systems in Section 2. The reduction of a conditional BU equational specification over Σ to a conditional equational specification over an expansion of Σ is proved for initial models.

In Section 8 we investigate the relationship between *computability* and *algebraic specifiability* of functions on initial N-standard algebras, and prove Theorem D. Finally, in Section 9, we consider the converse problem of finding sufficient conditions for algebraic specifiability to imply computability on classes of standard structures. Two equivalence theorems are proved.

We wish to thank an anonymous referee for some very helpful comments.

1   Many-sorted signatures and algebras

In this section we briefly review concepts defined and discussed in [TZ00, §1], where more detailed information can be found. Background information on universal algebra can be found in [MT92, EM85, Wec92].

1.1   Basic definitions

A *signature* Σ (for a many-sorted algebra) is a pair consisting of (i) a finite set $\boldsymbol{Sort}(\Sigma)$ of *sorts*, and (ii) a finite set $\boldsymbol{Func}(\Sigma)$ of *(primitive) function symbols*, each symbol F having a *type* $s_1 \times \cdots \times s_m \to s$, where $s_1, \ldots, s_m, s \in \boldsymbol{Sort}(\Sigma)$; in that case we write $F : s_1 \times \cdots \times s_m \to s$, with $\boldsymbol{dom}(F) =_{df} s_1 \times \cdots \times s_m$. (The case $m = 0$ corresponds to *constant symbols*.)

A Σ-*product type* has the form $u = s_1 \times \cdots \times s_m$ $(m \geq 0)$, where $s_1, \ldots, s_m$ are Σ-sorts. We use the notation $u, v, w, \ldots$ for Σ-product types.

A Σ-*algebra* A has, for each sort s of Σ, a non-empty *carrier set* A_s of sort s, and for each Σ-function symbol $F : u \to s$, a function $F^A : A^u \to A_s$ (where, for the Σ-product type $u = s_1 \times \cdots \times s_m$, we write $A^u =_{df} A_{s_1} \times \cdots \times A_{s_m}$).

Given an algebra A, we sometimes write $\Sigma(A)$ for its signature.

The algebra A is *total* if F^A is total for each Σ-function symbol F. Without such a totality assumption, A is called *partial*.

In this paper we deal with total algebras, except in §8.4.

We will also consider classes $\mathbb{K}$ of Σ-algebras. A Σ-*adt (abstract data type)* is defined to be any such class, closed under Σ-isomorphism. In particular, $\boldsymbol{Alg}(\Sigma)$ denotes the class of all Σ-algebras.

Examples. (*a*) The algebra of *booleans* has the carrier $\mathbb{B} = \{\mathtt{tt}, \mathtt{ff}\}$ of sort bool. It can be displayed as follows:

algebra	$\mathcal{B}$
carriers	$\mathbb{B}$
functions	$\mathtt{tt}, \mathtt{ff} : \to \mathbb{B},$
	$\mathsf{and}^{\mathcal{B}}, \mathsf{or}^{\mathcal{B}} : \mathbb{B}^2 \to \mathbb{B}$
	$\mathsf{not}^{\mathcal{B}} : \mathbb{B} \to \mathbb{B}$
end	

with signature

signature	$\Sigma(\mathcal{B})$
sorts	bool
functions	$\mathsf{true}, \mathsf{false} : \to \mathsf{bool},$
	$\mathsf{and}, \mathsf{or} : \mathsf{bool}^2 \to \mathsf{bool}$
	$\mathsf{not} : \mathsf{bool} \to \mathsf{bool}$
end	

For notational simplicity, we will usually not distinguish between function names in the signature (true, etc.) and their intended interpretations ($\mathsf{true}^{\mathcal{B}} = \mathtt{tt}$, etc.)

(*b*) The algebra $\mathcal{N}_0$ of naturals has a carrier $\mathbb{N}$ of sort nat, together with the zero constant and successor function:

algebra	$\mathcal{N}_0$
carriers	$\mathbb{N}$
functions	$0 : \to \mathbb{N},$
	$\mathsf{S} : \mathbb{N} \to \mathbb{N}$
end	

(*c*) The ring $\mathcal{R}_0$ of reals has a carrier $\mathbb{R}$ of sort real:

algebra	$\mathcal{R}_0$
carriers	$\mathbb{R}$
functions	$0, 1 : \to \mathbb{R},$
	$+, \times : \mathbb{R}^2 \to \mathbb{R},$
	$- : \mathbb{R} \to \mathbb{R}$
end	

We make the following assumption about the signatures Σ.

Instantiation Assumption. *For every sort s of Σ, there is a closed term of that sort, called the default term $\boldsymbol{\delta}^s$ of that sort.*

This guarantees the presence of *default values* $\boldsymbol{\delta}_A^s$ in a Σ-algebra A at all sorts s, and *default tuples* $\boldsymbol{\delta}_A^u$ at all product types u.

1.2 Some definitions

Definition 1 (Subalgebra). Given Σ-algebras A and B, we say that B is a Σ-*subalgebra* of A (written $B \preceq A$) iff (i) for all Σ-sorts s, $B_s \subseteq A_s$, and (ii) for every Σ-function symbol F, $F^B = F^A \restriction B$.

Definition 2 (Expansions and reducts). Let Σ and Σ' be signatures with $\Sigma \subset \Sigma'$.

(a) If A' is a Σ'-algebra, then the Σ-*reduct of A'*, $A' \mid_{\Sigma}$, is the algebra of signature Σ, consisting of the carriers of A' named by the sorts of Σ and equipped with the functions of A' named by the function symbols of Σ.

(b) If A is a Σ-algebra and A' is a Σ'-algebra, then A' is a Σ'-*expansion* of A iff A is the Σ-reduct of A'.

(c) If $\mathbb{K}'$ is a Σ'-adt, then $\mathbb{K}' \mid_{\Sigma}$ is the class of Σ-reducts of algebras in $\mathbb{K}'$.

1.3 Adding booleans: Standard signatures and algebras

Recall the algebra $\mathcal{B}$ of booleans (Example (a) in §1.1).

A signature Σ is called *standard* if (i) $\Sigma(\mathcal{B}) \subseteq \Sigma$; ($ii$) the Σ-function symbols include a *conditional*

$$\mathsf{if}_s : \mathsf{bool} \times s^2 \to s$$

for all sorts s of Σ other than bool; and (iii) the Σ-function symbols include an *equality operation*

$$\mathsf{eq}_s : s^2 \to \mathsf{bool}$$

for all $s \in \boldsymbol{EqSort}(\Sigma)$, where $\boldsymbol{EqSort}(\Sigma) \subseteq \boldsymbol{Sort}(\Sigma)$ is the set of Σ-*equality sorts*.

Given a standard signature Σ, a Σ-algebra A is *standard* if (i) it is an expansion of $\mathcal{B}$; (ii) the conditionals have their standard interpretation in A, *i.e.*, for $b \in \mathbb{B}$ and $x, y \in A_s$,

$$\mathsf{if}_s(b, x, y) \;=\; \begin{cases} x & \text{if } b = \mathtt{tt} \\ y & \text{if } b = \mathtt{ff}; \end{cases}$$

and (iii) the equality operator eq_s is interpreted as *identity* on each Σ-equality sort s.

Note that any many-sorted signature Σ can be *standardised* to a signature $\Sigma^{\mathcal{B}}$ by adjoining the sort bool together with the standard boolean operations; and, correspondingly, any algebra A can be standardised to an algebra $A^{\mathcal{B}}$ by adjoining the algebra $\mathcal{B}$ and the

conditional if_s at all Σ-sorts s, and the equality operator eq_s at the specified equality sorts:

```
algebra   A^B
import    A, B
          if_s : B × A_s^2 → A_s    (s ∈ Sort(Σ)),
          eq_s : A_s^2 → B          (s ∈ EqSort(Σ))
end
```

Thus the standardisation of a Σ-algebra A depends on the specification of $\boldsymbol{EqSort}(\Sigma)$. These will be the sorts for which an equality test is considered to be "computable" in some sense.

Examples. (a) The simplest standard algebra is the algebra $\mathcal{B}$ of the booleans.

(b) The standard algebra of naturals $\mathcal{N}$ is formed by standardising the algebra $\mathcal{N}_0$ (Example (b) in §1.1) with nat as an equality sort, and, further, adjoining the order relation $\mathsf{less}_{\mathsf{nat}}$ as a boolean-valued operation on $\mathbb{N}$:

```
algebra     N
import      N_0, B
functions   if_nat : B × N^2 → N,
            eq_nat, less_nat : N^2 → B
end
```

(c) The standard algebra $\mathcal{R}$ of reals is formed similarly by standardising the ring $\mathcal{R}_0$ (Example (c) in §1.1), with real *not* an equality sort. In fact, neither the equality nor the order relation on $\mathbb{R}$ is included as an operation on real. (The significance of this is discussed later; *cf.* Remark 3 in §5.3.)

$\boldsymbol{StdAlg}(\Sigma)$ denotes the class of all standard Σ-algebras.

1.4 Adding counters: N-standard signatures and algebras

A standard signature Σ is called *N-standard* if it includes (as well as bool) the *numerical sort* nat, and also function symbols for the *standard operations* of *zero* and *successor*, as well as the *conditional* and *equality* and *order* on the naturals:

$$0 : \quad \to \mathsf{nat}$$
$$\mathsf{S} : \mathsf{nat} \to \mathsf{nat}$$
$$\mathsf{if}_{\mathsf{nat}} : \mathsf{bool} \times \mathsf{nat} \to \mathsf{nat}$$
$$\mathsf{eq}_{\mathsf{nat}} : \mathsf{nat}^2 \to \mathsf{bool}$$
$$\mathsf{less}_{\mathsf{nat}} : \mathsf{nat}^2 \to \mathsf{bool}.$$

The corresponding Σ-algebra A is *N-standard* if the carrier A_{nat} is the set of natural numbers $\mathbb{N} = \{0,1,2, \dots \}$, and the standard operations (listed above) have their *standard interpretations* on $\mathbb{N}$.

Note that any standard signature Σ can be N-*standardised* to a signature Σ^N by adjoining the sort nat and the operations 0, S, eq_{nat}, $less_{nat}$ and if_{nat}. Correspondingly, any standard Σ-algebra A can be N-*standardised* to an algebra A^N by adjoining the carrier $\mathbb{N}$ together with the corresponding standard functions:

$$
\boxed{
\begin{array}{ll}
\text{algebra} & A^N \\
\text{import} & A, \mathcal{N} \\
\text{end} &
\end{array}
}
$$

Examples. (*a*) The simplest N-standard algebra is $\mathcal{N}$ (Example (*b*) in §1.3).

(*b*) The N-standard algebra $\mathcal{R}^N$ of reals is formed by N-standardising the standard real algebra $\mathcal{R}$ (Example (*c*) in §1.3).

NStdAlg (Σ) denotes the class of all N-standard Σ-algebras.

N-standardness Assumption. *We will assume throughout this paper that the signatures and algebras are N-standard, except where stated otherwise.*

We also consider a notion stricter than N-standardness.

1.5 Strictly N-standard signatures and algebras

An N-standard signature Σ is *strictly N-standard* if the *only operations of* Σ with range sort nat or bool are the *standard numerical operations* 0, S, if_{nat} eq_{nat}, $less_{nat}$ (§1.4) and the *boolean operations* true, false, and, or, not (§1.1). An algebra is *strictly N-standard* if its signature is.

Remarks.

(1) Any N-standardised signature and algebra are automatically strictly N-standard.

(2) A strictly N-standard signature has no equality sorts other than nat.

(3) Any subterm of a term of sort nat or bool of a strictly standard signature is itself of sort nat or bool. (Proved by structural induction on the term.)

The notion of strict N-standardness will be used in Section 9.

1.6 Adding arrays: Algebras A^* of signature Σ^*

The significance of arrays for computation is that they provide *finite but unbounded memory*.

Given a standard signature Σ, and standard Σ-algebra A, we expand Σ and A in two stages:

(1°) N-standardise these to form Σ^N and A^N, as in §1.3.

(2°) Define, for each sort s of Σ, the carrier A_s^* to be the set of *finite sequences* or *arrays* a^* over A_s, of "starred sort" s^*.

The reason for introducing starred sorts is the lack of effective coding of finite sequences within abstract algebras in general. (Note that, for simplicity, our definition excludes a starred sort nat^*, which would be redundant.)

The resulting algebras A^* have signature Σ^*, which expands Σ^N by including, for each sort s of Σ, the new starred sort s^*, and also the following new function symbols:

(i) the operator $\mathsf{Lgth}_s : s^* \to \mathsf{nat}$, where $\mathsf{Lgth}(a^*)$ is the length of the array a^*;

(ii) the application operator $\mathsf{Ap}_s : s^* \times \mathsf{nat} \to s$, where

$$
\mathsf{Ap}_s^A(a^*, k) \quad \left\{ \begin{array}{ll} a^*[k] & \text{if} \ \ k < \mathsf{Lgth}(a^*) \\ \boldsymbol{\delta}^s & \text{otherwise} \end{array} \right.
$$

where $\boldsymbol{\delta}^s$ is the default value at sort s guaranteed by the Instantiation Assumption ($\S1.1$)[1];

(iii) the null array $\mathsf{Null}_s : s^*$ of zero length;

(iv) the operator $\mathsf{Update}_s : s^* \times \mathsf{nat} \times s \to s^*$, where $\mathsf{Update}_s^A(a^*, n, x)$ is the array $b^* \in A_s^*$ of length $\mathsf{Lgth}(b^*) = \mathsf{Lgth}(a^*)$, such that for all $k < \mathsf{Lgth}(a^*)$

$$
b^*[k] \ = \ \left\{ \begin{array}{ll} a^*[k] & \text{if} \ \ k \neq n \\ x & \text{if} \ \ k = n \end{array} \right.
$$

(v) the operator $\mathsf{Newlength}_s : s^* \times \mathsf{nat} \to s^*$, where $\mathsf{Newlength}_s^A(a^*, m)$ is the array b^* of length m such that for all $k < m$,

$$
b^*[k] \ = \ \left\{ \begin{array}{ll} a^*[k] & \text{if} \ \ k < \mathsf{Lgth}(a^*) \\ \boldsymbol{\delta}^s & \text{if} \ \ \mathsf{Lgth}(a^*) \leq k < m \end{array} \right.
$$

(vi) the *conditional* on A_s^* for each sort s; and

(vii) the *equality* operator on A_s^* for each equality sort s.

Note that A^* is an N-standard Σ^*-*expansion* of A.

The justification for (vii) is that if a sort s has "computable" equality, then clearly so has the sort s^*, since it amounts to testing equality of finitely many pairs of objects of sort s, up to a computable length.

2 Proof systems and theories for Σ-algebras

To reason about computations, we choose a first-order language based on Σ as a specification language.

Note, in this connection, that the *operations* in Σ are used for computation. In particular, boolean-valued operations are used for tests in computations. By contrast, for specification and reasoning about these algebras, we may add *predicates* to the language,

---

[1] We assume that $a^*[k]$ is undefined for $k \geq \mathsf{Lgth}(a^*)$.

which are not, in general, computable or testable. For example, our specification language will include the equality predicate at all sorts (as we will see), whereas only the *equality sorts s* have the "computable" equality operator eq_s (§1.3). In writing specifications on the reals we may also add the '<' predicate (again, not computable, at least if defined totally), as we will do later (§5.3) for the specification of approximable computability. Note that these predicates added to the language do not form part of the signature. Intuitively, think of the equality operation as a "computable" boolean test, but the equality predicate as a "provable" assertion of equality between two terms.

So let $\boldsymbol{Lang}(\Sigma)$ be the first order language over the signature Σ, with the *equality predicate* at all sorts. The syntax of $\boldsymbol{Lang}(\Sigma)$ is generated as follows. For each Σ-sort s there are countably many *variables* of sort s, denoted a, b, ..., x, y, ... Next, for each Σ-sort s, there are *terms* of sort s, generated from variables and the function symbols of Σ according to the standard typing rules. We write t^s or $t : s$ if t is a term of sort s, and, for a product type $u = s_1 \times \cdots \times s_m$, we write $t : u$ if t is a u-tuple of terms, *i.e.*, an m-tuple of terms of type $s_1 \times \cdots \times s_m$.

The *atomic formulae* of $\boldsymbol{Lang}(\Sigma)$, then, are equations $(t_1^s = t_2^s)$ between terms of sort s, for all Σ-sorts s (whether equality sorts or not), and the propositional constants true and false. *Formulae* of $\boldsymbol{Lang}(\Sigma)$ are built up from these by the logical connectives $\wedge$, $\vee$, $\rightarrow$, $\neg$, and the quantifiers $\forall_s$ and $\exists_s$ for all sorts s of Σ.

We will consider (in the following four subsections) four formal systems in $\boldsymbol{Lang}(\Sigma)$, conveniently formulated as *sequent calculi*. The first is our basic system $\mathsf{FOL}(\Sigma)$, full first order logic with equality over Σ. The next two are subsystems of this, which will be used in Section 7. The final system is a subsystem of $\mathsf{FOL}(\Sigma)$, extended by an infinitary proof rule.

Background information on sequent calculus proof systems can be found in [Tak87].

Note that we do *not* assume (N-)standardness of Σ in subsections 2.1 and 2.2 (only) below.

2.1 FOL(Σ): Full first order logic with equality over Σ

This can be formulated in a system $\mathsf{LK_e}(\Sigma)$, which is an adaptation to the many-sorted signature Σ of the systems LK and $\mathsf{LK_e}$ of [Gen69, Tak87]. The atomic formulae are equations at all Σ-sorts.

A *sequent* of $\mathsf{LK_e}(\Sigma)$ is a construct of the form $\Gamma \longmapsto \Delta$, where Γ and Δ are each finite sequences of formulae of $\boldsymbol{Lang}(\Sigma)$.

Derivations (of sequents) are then constructed from certain specified *initial sequents* ("axioms") by means of specified *inference rules*.

The system $\mathsf{LK_e}$ can be augmented in two ways:

(*a*) Adding *axioms of a theory*, or rather all substitution instances of these, as initial sequents;

(*b*) Adding *induction* for a class $\mathcal{C}$ of formulae (in case Σ is N-standard), in the form of

the inference rule

$$\mathcal{C}\text{-}\mathsf{Ind}(\Sigma) : \quad \frac{\Gamma \longmapsto \Delta, F(0) \qquad F(\mathsf{a}), \Pi \longmapsto \Lambda, F(\mathsf{Sa})}{\Gamma, \Pi \longmapsto \Delta, \Lambda, F(t)}$$

where the induction variable a has sort nat, and the induction formula $F(\mathsf{a})$ belongs to the class $\mathcal{C}$. We write $\mathsf{Ind}(\Sigma)$ for full Σ-induction, *i.e.*, where $\mathcal{C}$ is the set of all first-order Σ-formulae.

We will also be interested in the "intuitionistic" version $\mathcal{C}\text{-}\mathsf{Ind}_i$ of $\mathcal{C}\text{-}\mathsf{Ind}$, in which the sequences Δ and Λ above are empty.

Analogous augmentations can be made for the other systems considered below.

In the next three subsections we will consider three further systems, the first two of which are subsystems of $\mathsf{FOL}(\Sigma)$ and the third of which is a subsystem of $\mathsf{FOL}(\Sigma)$ augmented by an infinitary ω-rule. These subsystems are, in fact, also subsystems of $\mathsf{LJ_e}(\Sigma)$, which is an adaptation to Σ of the "intuitionistic" system $\mathsf{LJ_e}$ (*loc. cit.*), in which the sequents have only one formula on the rhs. (When we are working with these subsystems, the scheme $\mathcal{C}\text{-}\mathsf{Ind}$ will consist of intuitionistic sequents, so that the sequences Δ and Λ above are empty.)

2.2 $\mathsf{CondEq}(\Sigma)$: conditional equational logic over Σ

A *conditional equation* is a formula of the form

$$P_1 \wedge \ldots \wedge P_n \to P \tag{$*$}$$

where $n \geq 0$ and P_i and P are equations. A *conditional equational theory* is a set of such formulae (or their universal closures). An *equational sequent* is a sequent of the form

$$P_1, \ldots, P_n \longmapsto P$$

where $n \geq 0$ and P_i and P are equations. This sequent *corresponds to* the conditional equation $(*)$.

The *initial sequents* are all substitution instances of the Σ-equality axioms (expressing that equality is a congruence relation with respect to Σ), and the *inferences* are *structural inferences*, *atomic cuts* and *substitution* of terms for free variables in sequents.

2.3 $\mathsf{CondBUEq}(\Sigma)$: Conditional BU equational logic over Σ

A *BU (bounded universal) quantifier* is a quantifier of the form '$\forall \mathsf{z} < t$', where $\mathsf{z} : \mathsf{nat}$ and $t : \mathsf{nat}$. (The most elegant approach is to think of this as a primitive construct, with its own introduction rule: see below.) A $(\Sigma\text{-})BU$ *equation* is formed by prefixing an equation by a string of 0 or more bounded universal quantifiers. A *conditional BU equation* is a formula of the form

$$Q_1 \wedge \ldots \wedge Q_n \to Q \tag{$**$}$$

where $n \geq 0$ and Q_i and Q are BU equations. A *conditional BU equational theory* is a set of such formulae (or their universal closures). A *BU equational sequent* is a sequent of the form

$$Q_1, \ldots, Q_n \longmapsto Q$$

where $n \geq 0$ and Q_i and Q are BU equations. This sequent *corresponds to* the conditional BU equation $(**)$.

The system $\mathsf{CondBUEq}(\Sigma)$ consists of *BU equational sequents*. The initial sequents are the Σ-equality axioms, as before, plus the *boundedness axioms*

$$\mathsf{BddAx}(\Sigma): \qquad\qquad P(0), \ldots, P(\overline{n-1}) \longmapsto \forall \mathsf{z} < \bar{k} P(\mathsf{z})$$

for all Σ-equations P and all $n \in \mathbb{N}$, where $\bar{n}$ is the *numeral* for n, *i.e.*, the term $\mathsf{S} \ldots \mathsf{S}0$ (n times 'S'). The only inferences are *structural inferences*, *cut*, *substitution*, and the rules for the BU quantifiers:

$$\forall_b L : \; \frac{\Gamma \longmapsto s < t \qquad Q(s), \Delta \longmapsto Q}{\forall \mathsf{z} < t Q(\mathsf{z}), \Gamma, \Delta \longmapsto Q} \qquad\qquad \forall_b R : \; \frac{\mathsf{a} < t, \Gamma \longmapsto Q(\mathsf{a})}{\Gamma \longmapsto \forall \mathsf{z} < t Q(\mathsf{z})}$$

where s and t are terms of sort nat, '$s < t$' stands for '$\mathsf{less_{nat}}(s, t) = \mathsf{true}$', and the variable $\mathsf{a} : \mathsf{nat}$ is the '*eigenvariable*' of the inference $\forall_b R$, which does not occur in the conclusion of that inference.

Remark (Boundedness axioms). The boundedness axioms $\mathsf{BddAx}(\Sigma)$ hold (of course) in N-standard models. We remark here that they are derivable in $\mathsf{FOL}(\Sigma)$ from the *N-standardness axioms* $\mathsf{NStdAx}_0(\Sigma)$ (a set of conditional equations defined in §7.2), plus the single formula

$$\mathsf{z}_1 < \mathsf{S}\mathsf{z}_2 \; \longrightarrow \; \mathsf{z}_1 < \mathsf{z}_2 \vee \mathsf{z}_1 = \mathsf{z}_2$$

which is, however, *not a conditional BU equation*. This formula is derivable, in turn, in $\mathsf{FOL}(\Sigma) + \mathsf{QF\text{-}Ind}(\Sigma)$ (induction for quantifier-free formulae), from $\mathsf{NStdAx}_0(\Sigma)$. It is not clear whether the boundedness axioms are derivable in conditional BU equational logic alone from $\mathsf{NStdAx}_0(\Sigma)$, which is why we are adding them as axioms.

2.4 $\mathsf{CondSUEq}_\omega(\Sigma)$:   Conditional SU equational logic over Σ

The final two systems that interest us, in this and the next subsection, are not subsystems of $\mathsf{LK_e}$, but *infinitary* systems. They will be used for another illustration of a Malcev-type theorem for N-standard algebras (see Section 6, Theorem 4). However they will not be used in the investigation of the relationship between computability and algebraic specifiability in Section 8.

A $(\Sigma\text{-})SU$ *(standard universal) equation* is formed by prefixing an equation by a string of 0 or more universal quantifiers of sort nat. A *conditional SU equation* is a formula of the form

$$R_1 \wedge \ldots \wedge R_n \to R \qquad\qquad\qquad\qquad (***)$$

where $n \geq 0$ and R_i and R are SU equations. A *conditional SU equational theory* is a set of such formulae (or their universal closures). An *SU equational sequent* is a sequent of the form

$$R_1, \ldots, R_n \longmapsto R$$

where $n \geq 0$ and R_i and R are SU equations. This sequent *corresponds to* the conditional SU equation $(***)$.

The system $\mathsf{CondSUEq}_\omega(\Sigma)$ contains SU equational sequents. It contains the equality axioms and the following inferences: the structural inferences, cut, and the following rules for the universal number quantifier (where $t : \mathsf{nat}$):

$$\forall L : \quad \frac{R(t), \Gamma \longmapsto R}{\forall \mathsf{z} R(\mathsf{z}), \Gamma \longmapsto R} \qquad\qquad \forall_\omega R : \quad \frac{\ldots\; \Gamma \longmapsto R(\bar{n})\;\ldots\quad (\text{all } n \in \mathbb{N})}{\Gamma \longmapsto \forall \mathsf{z} R(\mathsf{z})} \qquad (*)$$

Note that the rule $\forall_\omega R$ is actually an *infinitary ω-rule*.

2.5 $\mathsf{FOL}_\omega(\Sigma)$: full first-order logic with equality and an ω-rule over Σ

This modifies the system $\mathsf{FOL}(\Sigma)$ (§2.1) by replacing the usual universal number quantifier rule $\forall R$ by the infinitary rule $\forall_\omega R$ (§2.4), with also the corresponding rule $\exists_\omega L$ dually. We omit details, except to point out that $\mathsf{FOL} + \mathsf{Ind}(\Sigma)$ can easily be interpreted in it.

We write $\boldsymbol{Eq}(\Sigma)$, $\boldsymbol{BUEq}(\Sigma)$ and $\boldsymbol{SUEq}(\Sigma)$ for the classes of equations, BU equations and SU equations (respectively) over Σ.

2.6 Conservativity lemmas

One reason for the importance of (finite or infinite) conditional equational logic lies in the following lemmas. First we need a definition which will be given again in context in Section 7. Let $\mathcal{F}$ be a formal system (typically $\mathsf{CondEq}(\Sigma)$ or $\mathsf{CondEq}_\omega(\Sigma)$), and let T be a theory over Σ (typically a conditional equational or ω-conditional equational theory). We say that T *determines* nat in $\mathcal{F}$ if every closed term of sort nat is, provably in $\mathcal{F}$ from T, equal to a numeral.

(1) (FOL over CondEq.) Let E be a Σ-conditional equational theory, and let $\Gamma \longmapsto P$ be a Σ-equational sequent. Then $\Gamma \longmapsto P$ is provable from E in $\mathsf{FOL}(\Sigma)$ if, and only if, it is provable from E in $\mathsf{CondEq}(\Sigma)$.

(2) ($\mathsf{FOL} + \mathsf{Ind}$ over CondEq.) Let E be a Σ-conditional equational theory which determines nat in $\mathsf{CondEq}(\Sigma)$, and let $\Gamma \longmapsto P$ be a *closed* Σ-equational sequent. Then $\Gamma \longmapsto P$ is provable from E in $\mathsf{FOL}(\Sigma) + \mathsf{Ind}(\Sigma)$ if, and only if, it is provable from E in $\mathsf{CondEq}(\Sigma)$.

(3) ($\mathsf{FOL} + \mathsf{Ind}$ over $\mathsf{CondBUEq}$.) Let F be a Σ-conditional BU equational theory which determines nat in $\mathsf{CondBUEq}(\Sigma)$, and let $\Gamma \longmapsto Q$ be a *closed* Σ-BU equational sequent. Then $\Gamma \longmapsto Q$ is provable from F in $\mathsf{FOL}(\Sigma) + \mathsf{Ind}(\Sigma)$ if, and only if, it is provable from F in $\mathsf{CondBUEq}(\Sigma)$.

(4) (FOL_ω over $\mathsf{CondSUEq}_\omega$.) Let G be a Σ-conditional SU equational theory over Σ which determines nat in $\mathsf{CondSUEq}_\omega(\Sigma)$, and let $\Gamma \longmapsto R$ be a *closed* Σ-conditional SU

equational sequent. Then $\Gamma \longmapsto R$ is provable from G in $\mathsf{FOL}_\omega(\Sigma)$ if, and only if, it is provable from G in $\mathsf{CondSUEq}_\omega(\Sigma)$.

All four lemmas can be proved by cut elimination. We omit proofs, except to note briefly that the two conditions, that E determines nat and that $\Gamma \longmapsto P$ is closed, are used in (2) and (3) to eliminate induction inferences, and in (4) to eliminate cuts of formulae universally or existentially quantified over nat.

Remarks. (1) These conservativity lemmas (at least for simple equations) also follow from the Birkhoff-Mal'cev-type completeness theorems 1–4 in Section 7.

(2) Infinitary systems come into their own when reasoning about infinite objects such as infinite streams of data. Some applications in this direction, using a related infinitary system (CondEq_ω), are given in [TZ01b].

3 Specifiability of functions by theories

3.1 Specifiability over algebras and over classes of algebras

Recall from Section 2 that $\boldsymbol{Lang}(\Sigma)$ is the first order language over Σ, with equality as the only predicate at all sorts.

A Σ-*theory* is just a set T of formulae in $\boldsymbol{Lang}(\Sigma)$. The *axioms* of T are the formulae in T. We will be particularly interested in theories T satisfying certain syntactic conditions; for example, T might be a set of conditional equations. This is considered more carefully in Section 7.

We are also interested (when Σ is N-standard) in classes $\mathbb{K}$ of the *N-standard models* of such Σ-theories: $\mathbb{K} = \boldsymbol{NStdAlg}(\Sigma, T) \subseteq \boldsymbol{NStdAlg}(\Sigma)$. In this case we say also that (Σ, T) is an (N-standard) *specification* for the adt $\mathbb{K}$.

Assume, for the rest of this section, that Σ, Σ' and Σ'' are N-standard signatures with $\Sigma \subset \Sigma' \subset \Sigma''$. Also, A is an N-standard Σ-algebra and A' is an N-standard Σ'-algebra. Also, T is a Σ-theory, T' is a Σ'-theory and T'' is a Σ''-theory.

Note that any expansion of a standard algebra is also standard, and any expansion of an N-standard algebra is also N-standard.

Definition 1. Let A'_1 and A'_2 be two Σ'-algebras with $A'_1 \restriction_\Sigma = A'_2 \restriction_\Sigma$. Then A'_1 and A'_2 are Σ'/Σ-*isomorphic*, written $A'_1 \cong_{\Sigma'/\Sigma} A'_2$, if there is a Σ'-isomorphism from A'_1 to A'_2 whose restriction to Σ is the identity on $A'_1 \restriction_\Sigma$.

Definition 2. Suppose A' is a Σ'-expansion of A. We say that (Σ', T') *specifies A' over A* iff A' is the unique (up to Σ'/Σ-isomorphism) Σ'-expansion of A satisfying T'; in other words:

(i) $A' \models T'$; and

(ii) for all Σ'-expansions B' of A, if $B' \models T'$ then $B' \cong_{\Sigma'/\Sigma} A'$.

We will occasionally write: "T' specifies A' over A" instead of "(Σ', T') specifies A' over A".

An important special case of Definition 2 is the following.

Definition 2$^{\boldsymbol{f}}$. Suppose $\Sigma' = \Sigma \cup \{\, \mathsf{f} \,\}$. We say that (Σ', T') *specifies f over A* iff f is the unique (up to Σ'/Σ-isomorphism) function on A (of the type of f) such that $(A, f) \models T'$.

Definition 3. Suppose A' is a Σ'-expansion of A. We say that (Σ'', T'') *specifies A' over A with hidden sorts and/or functions* iff A' is the unique (up to Σ'/Σ-isomorphism) Σ'-expansion of A such that some Σ''-expansion of A' satisfies T''; in other words:

(i) A' is a Σ'-reduct of a Σ''-model of T''; and

(ii) for all Σ'-expansions B' of A, if B' is a Σ'-reduct of a standard Σ''-model of T'', then $B' \cong_{\Sigma'/\Sigma} A'$.

Again, an important special case:

Definition 3$^{\boldsymbol{f}}$. Suppose $\Sigma' = \Sigma \cup \{\, \mathsf{f} \,\}$. We say that (Σ'', T'') *specifies f over A with hidden sorts and/or functions* iff f is the unique function on A (of the type of f) such that some Σ''-expansion of (A, f) satisfies T''.

Definition 4. An operator $\Phi : \boldsymbol{NStdAlg}\,(\Sigma) \to \boldsymbol{NStdAlg}(\Sigma')$ is *expanding (over Σ)* iff for all N-standard Σ-algebras A, $\Phi(A)$ is a Σ'-expansion of A, *i.e.*, $\Phi(A)\,|\,_{\Sigma} = A$.

Example. The array construction $A \mapsto A^{*}$ is an expanding operator.

Assume further, for the rest of this section, that $\Phi : \boldsymbol{NStdAlg}\,(\Sigma) \to \boldsymbol{NStdAlg}(\Sigma')$ *is an expanding operator over Σ, and that* $\mathbb{K} \subseteq \boldsymbol{NStdAlg}\,(\Sigma)$.

Notation. (1) We will write A^{Φ} for $\Phi(A)$.

(2) We write $\mathbb{K}^{\Phi}$ for (the closure w.r.t. Σ'-isomorphism of) the class $\{A^{\Phi} \mid A \in \mathbb{K}\} \subseteq \boldsymbol{NStdAlg}(\Sigma')$.

Definition 5.

(a) (Σ', T') *specifies Φ uniformly over $\mathbb{K}$* iff for all $A \in \mathbb{K}$, (Σ', T') specifies A^{Φ} over A.

(b) (Σ', T') *specifies Φ uniformly over Σ* iff (Σ', T') specifies Φ uniformly over $\boldsymbol{NStdAlg}\,(\Sigma)$.

Proposition 1. *Suppose (Σ', T') specifies Φ uniformly over $\mathbb{K}$.*

(i) For $A \in \mathbb{K}$, $A \models T \iff A^{\Phi} \models T + T'$.

(ii) If $\mathbb{K} = \boldsymbol{NStdAlg}(\Sigma, T)$, then $\mathbb{K}^{\Phi} = \boldsymbol{NStdAlg}(\Sigma', T + T')$.

Definition 6.

(a) (Σ'', T'') *specifies* Φ *uniformly over* $\mathbb{K}$ *with hidden sorts and/or functions* iff for all $A \in \mathbb{K}$, (Σ'', T'') specifies A^{Φ} over A with hidden sorts and/or functions.

(b) (Σ'', T'') *specifies* Φ *uniformly over* Σ *with hidden sorts and/or functions* iff (Σ'', T'') specifies Φ uniformly over **NStdAlg**(Σ) with hidden sorts and/or functions.

Proposition 2. *Suppose* (Σ'', T'') *specifies* Φ *uniformly over* Σ *with hidden sorts and/or functions.*

(i) $A \models T \Longleftrightarrow A^{\Phi}$ *is a* Σ'*-retract of a* Σ''*-model of* $T + T''$.

(ii) *If* $\mathbb{K} = $ **NStdAlg**(Σ, T), *then* $\mathbb{K}^{\Phi} = \big($**NStdAlg**$(\Sigma'', T + T'')\big) \mid _{\Sigma'}$.

Interesting special cases of the above notions, in which the theories T, T' and T'' are subject to certain syntactic conditions, are considered below (§3.3) and in Section 7. First we give an important example of a specification of an expanding operator.

We write *conditional equational specification* and *conditional BU equational specification* for specifications in which the formulae are all conditional equations and conditional BU equations, respectively.

3.2 Conditional BU equational specification of the array construction

Let $\mathsf{ArrAx}(\Sigma)$ be the following set of axioms in A (dropping sort subscripts):

$$
\begin{array}{c}
\mathsf{Lgth}(\mathsf{Null}) = 0, \\
\mathsf{less_{nat}}(\mathsf{z}, \mathsf{Lgth}(\mathsf{a})) = \mathsf{false} \quad \rightarrow \quad \mathsf{Ap}(\mathsf{a}, \mathsf{z}) = \boldsymbol{\delta}, \\
\mathsf{Lgth}(\mathsf{Update}(\mathsf{a}, \mathsf{z}, \mathsf{x})) = \mathsf{Lgth}(\mathsf{a}), \\
\mathsf{eq_{nat}}(\mathsf{z}, \mathsf{z_0}) = \mathsf{false} \quad \rightarrow \quad \mathsf{Ap}(\mathsf{Update}(\mathsf{a}, \mathsf{z_0}, \mathsf{x}), \mathsf{z}) = \mathsf{Ap}(\mathsf{a}, \mathsf{z}), \\
\mathsf{less_{nat}}(\mathsf{z}, \mathsf{Lgth}(\mathsf{a})) = \mathsf{true} \quad \rightarrow \quad \mathsf{Ap}(\mathsf{Update}(\mathsf{a}, \mathsf{z}, \mathsf{x}), \mathsf{z}) = \mathsf{x}, \\
\mathsf{Lgth}(\mathsf{Newlength}(\mathsf{a}, \mathsf{z})) = \mathsf{z}, \\
\mathsf{less_{nat}}(\mathsf{z}, \mathsf{z_1}) = \mathsf{true} \quad \rightarrow \quad \mathsf{Ap}(\mathsf{Newlength}(\mathsf{a}, \mathsf{z_1}), \mathsf{z}) = \mathsf{Ap}(\mathsf{a}, \mathsf{z}), \\
\mathsf{Lgth}(\mathsf{a_1}) = \mathsf{Lgth}(\mathsf{a_2}) \wedge \forall \mathsf{z} < \mathsf{Lgth}(\mathsf{a_1}) \big[\mathsf{Ap}(\mathsf{a_1}, \mathsf{z}) = \mathsf{Ap}(\mathsf{a_2}, \mathsf{z})\big] \quad \rightarrow \quad \mathsf{a_1} = \mathsf{a_2}.
\end{array}
$$

The last axiom relates equality on s^* to equality on s, for all equality sorts s except nat (since there is no starred sort nat^*, as explained in §1.6).

Note that all the axioms of $\mathsf{ArrAx}(\Sigma)$ are conditional equations, except for the last one, which is a conditional BU equation!

Theorem 1. *The specification* $(\Sigma^*, \mathsf{ArrAx}(\Sigma))$ *specifies the array construction* $A \mapsto A^*$ *uniformly over* Σ.

Proof (outline): Given an N-standard Σ-algebra A, and a Σ-sort s, the axioms for 'Null', 'Newlength' and 'Update' guarantee that *at least* all the "standard" arrays over A_s are present (or can be "constructed"). On the other hand, the axiom for array equality

guarantees that there are no "non-standard" arrays, *i.e.*, no elements of A_s^* other than these. $\square$

This array specification will be considered again, from the viewpoints of specification of μPR^* computations (§5.2), and initial algebra specifications (§8.2).

3.3   Reducing conditional BU to conditional equational specifications

Theorem 2 (BU elimination). *Let $\Sigma \subset \Sigma'$, let A' be a Σ'-expansion of A, and let F be a conditional BU equational Σ'-theory which specifies A' over A. Then there is an expansion Σ'' of Σ' by function symbols, and a conditional equational Σ''-theory E which specifies A' over A, with hidden functions. If F contains q occurrences of BU quantifiers, then Σ'' expands Σ' by q new function symbols. Moreover, if F is finite, with e axioms (say), then so is E, with $e + 4q$ axioms.*

Proof: The idea is to incorporate in the signature, for each BU quantifier occurring in F, a *characteristic function* for that quantifier, or (expressed differently) a function which computes that quantifier. Consider (for notational simplicity) the case of an equation with a single BU quantifier

$$\forall \mathsf{z} < s(\mathsf{x}) \big[t_1(\mathsf{z}, \mathsf{x}) = t_2(\mathsf{z}, \mathsf{x})\big]. \tag{$*$}$$

with $\mathsf{x} : u$. (In the general case, we "eliminate" the quantifiers successively, from the inside out.) We adjoin, for each such BU quantifier ($*$) occurring in F, a boolean-valued function symbol

$$\mathsf{f} : \ \mathsf{nat} \times u \ \rightarrow \ \mathsf{bool}$$

intended to satisfy in A

$$\mathsf{f}(n, x) = \mathsf{true} \ \iff \ \forall z < n \big[t_1(z, x) = t_2(z, x)\big].$$

for all $n \in \mathbb{N}$, $x \in A^u$. This interpretation is imposed on f by *adjoining* to F the following axioms giving the inductive definition for f:

$$\mathsf{f}(0, \mathsf{x}) = \mathsf{true}$$
$$\mathsf{f}(\mathsf{z}, \mathsf{x}) = \mathsf{true} \ \wedge t_1(\mathsf{z}, \mathsf{x}) = t_2(\mathsf{z}, \mathsf{x}) \ \rightarrow \ \mathsf{f}(\mathsf{Sz}, \mathsf{x}) = \mathsf{true}$$
$$\mathsf{f}(\mathsf{Sz}, \mathsf{x}) = \mathsf{true} \ \rightarrow \ \mathsf{f}(\mathsf{z}, \mathsf{x}) = \mathsf{true} \tag{$***$}$$
$$\mathsf{f}(\mathsf{Sz}, \mathsf{x}) = \mathsf{true} \ \rightarrow \ t_1(\mathsf{z}, \mathsf{x}) = t_2(\mathsf{z}, x)$$

and *replacing* ($*$) in F by

$$\mathsf{f}(s(\mathsf{x}), \mathsf{x}) = \mathsf{true}. \tag{$****$}$$

In this way we replace F by a conditional equational Σ''-theory E, with the stated properties. $\square$

Note that if F contains infinitely many occurences of BU quantifiers, then Σ'' contains, correspondingly, infinitely many new function symbols, which is (strictly speaking) not allowed by our definition of signature, although it is harmless enough here.

We will return to this topic in the context of initial algebra specifications in §7.7.

4   Computable functions

In this section we consider various notions of computability over abstract algebras. (An equivalent approach, using an imperative model of programming featuring the 'while' construct, was developed in [TZ88, TZ00] where the equivalence of these two approaches are explained.) In §4.1 two computability classes are introduced. In §4.2 two more classes are formed by adjoining the μ operator to these.

4.1   PR(Σ) and PR*(Σ) computable functions

Given an N-standard signature Σ, we define PR *schemes* over Σ which generalise the schemes for primitive recursive functions over $\mathbb{N}$ in [Kle52]. They define (total) functions f either outright (as in the base cases (i)—(ii) below) or from other functions $(g, \ldots, h, \ldots)$ (as in the inductive cases (iii)—(v)) as follows:

(a) Basic schemes: Initial functions

(i) *Primitive Σ-functions:*

$$f(x) \;=\; F(x)$$
$$f(x) \;=\; c$$

of type $u \to s$, for all the primitive function symbols $F \colon u \to s$ and constant symbols c of Σ, where $x : u$.

(ii) *Projection:*
$$f(x) \;=\; x_i$$

of type $u \to s_i$, where $x = (x_1, \ldots, x_m)$ is of type $u = s_1 \times \cdots \times s_m$.

(b) Inductive schemes:

(iii) *Composition:*
$$f(x) \;=\; h(g_1(x), \ldots, g_m(x))$$

of type $u \to s$, where $g_i \colon u \to s_i$ $(i = 1, \ldots, m)$ and $h \colon s_1 \times \cdots \times s_m \to s$.

(iv) *Definition by cases:*
$$f(b, x, y) \;=\; \begin{cases} x & \text{if } b = \mathsf{tt} \\ y & \text{if } b = \mathsf{ff} \end{cases}$$

of type $\mathsf{bool} \times s^2 \to s$.

(v) *Simultaneous primitive recursion on $\mathbb{N}$:* This defines, on each $A \in \boldsymbol{NStdAlg}\,(\Sigma)$, for fixed $m > 0$ (the degree of simultaneity), $n \geq 0$ (the number of parameters), and product types u and $v = s_1 \times \cdots \times s_m$, an m-tuple of functions $f = (f_1, \ldots, f_m)$ with $f_i : \mathsf{nat} \times u \to s_i$, such that for all $x \in A^u$ and $i = 1, \ldots, m$,

$$f_i(0, x) \;=\; g_i(x)$$
$$f_i(z + 1, x) \;=\; h_i(z, x, f_1(z, x), \ldots, f_m(z, x))$$

where $g_i \colon u \to s_i$ and $h_i \colon \mathsf{nat} \times u \times v \to s_1$ $(i = 1, \ldots, m)$.

Note that the last scheme uses the N-standardness of the algebras, *i.e.* the carrier $\mathbb{N}$.

For details of the syntax and semantics of PR schemes, see [TZ88, §4.1.5], from which it can be seen that a scheme for a function contains (hereditarily) the schemes for all the auxiliary functions used to define it.

In the context of algebraic specification theory, it often turns out to be more convenient to work with PR *derivations* instead of PR schemes. A PR derivation is, roughly, a "linear version" of a PR scheme, in which all the auxiliary functions are displayed in a list. More precisely:

Definition (PR derivation). A *PR(Σ) derivation* α is a list of pairs

$$\alpha \;=\; ((f_0, \sigma_0),\; (f_1, \sigma_1),\; \ldots\; (f_n, \sigma_n)) \tag{$*$}$$

of functions (actually function symbols) f_i and PR schemes σ_i $(i = 1, \ldots, n)$ where for each i, either f_i is an initial function, or f_i is defined by α_i from functions f_j, for certain $j < i$. The derivation α is said to be a *PR derivation of* f_n, with *auxiliary functions* $f_0, \ldots, f_{n-1}$. The *type* of α is the type of f_n.

Notation. A $\mathrm{PR}(\Sigma)_{u \to s}$ scheme (or derivation) is a $\mathrm{PR}(\Sigma)$ scheme (or derivation) of type $u \to s$.

Remarks. (1) The formalism of $\mathrm{PR}(\Sigma)$ derivations is equivalent to that of $\mathrm{PR}(\Sigma)$ schemes: from a PR scheme we can derive an equivalent PR derivation by "linearising" the subschemes, and conversely, given the derivation $(*)$, the scheme σ_n is equivalent to it. Below, we will usually work with derivations.

(2) A $\mathrm{PR}(\Sigma)_{u \to s}$ derivation $\alpha \colon u \to s$ defines, or rather computes, a function $\mathsf{f}_\alpha^A \colon A^u \to A_s$, or, more generally, a family of functions $\{\, \mathsf{f}_\alpha^A \mid A \in \boldsymbol{NStdAlg}\,(\Sigma)\,\}$ uniformly over $\boldsymbol{NStdAlg}\,(\Sigma)$.

(3) We assume a standard Gödel numbering of $\mathrm{PR}(\Sigma)$ derivations, writing $\ulcorner\alpha\urcorner$ for the Gödel number of derivation α.

It turns out that a broader class of functions provides a better generalisation of the notion of primitive recursiveness, namely *PR* computability*. A function on A is $\mathrm{PR}^*(\Sigma)$ computable if it is defined by a PR derivation over Σ^*, interpreted on A^* (*i.e.*, using starred sorts for the auxiliary functions used in its definition).

4.2 $\mu\mathbf{PR}(\Sigma)$ and $\mu\mathbf{PR}^*(\Sigma)$ computable functions

The μPR *schemes* over Σ are formed by adding to the PR schemes of §4.1 the inductive scheme:

(vi) Least number or μ *operator:*

$$f(x) \;\simeq\; \mu z[g(x, z) = \mathsf{tt}]$$

of type $u \to \mathsf{nat}$, where $g : u \times \mathsf{nat} \to \mathsf{bool}$ is μPR. Here $f(x) \downarrow z$ if, and only if, $g(x, y) \downarrow \mathsf{ff}$ for each $y < z$ and $g(x, z) \downarrow \mathsf{tt}$.

Note that this scheme also uses the N-standardness of the algebra. Also, μPR computable functions are, in general, *partial*. We use the notation $f(x) \downarrow y$ to mean that $f(x)$ is defined and equal to y. The notation '$\simeq$' means that the two sides are either both defined and equal, or both undefined. The schemes for composition and simultaneous primitive recursion are correspondingly re-interpreted to allow for partial functions.

These schemes generalise the schemes given in [Kle52] for partial recursive functions over $\mathbb{N}$.

As before, we can define the concepts of μPR(Σ) *derivations* and μPR(Σ) *computability*.

Again, a broader class turns out to be more useful, namely μPR^* *computability*. This is just PR* computability with μ.

Notation. PR(A) is the class of functions PR computable on A, and PR$(A)_{u \to s}$ is the subclass of PR(A) of functions of type $u \to s$. Similarly for PR$^*(A)$, μPR(A), etc.

There are many other models of computability, due to Moschovakis, Friedman, Shepherdson and others, which turn out to be equivalent to μPR* computability: see [TZ00, §7]. All these equivalences have led to the postulation of a *generalised Church-Turing Thesis for deterministic computation of functions*, which can be roughly formulated as follows:

> *Computability of functions on many-sorted algebras by deterministic algorithms can be formalised by μPR^* computability.*

4.3 Equivalent imperative programming models of computation

In [TZ00] we investigate computation on many-sorted Σ-algebras, using imperative programming models: **While**(Σ), based on the 'while' loop construct over Σ, **For**(Σ), based similarly on the 'for' loop, and **While**$^*(\Sigma)$ and **For**$^*(\Sigma)$, which use arrays, *i.e.*, auxiliary variables of starred sort over Σ.

Writing **While**(A) for the class of functions **While**-computable on A, etc., we can list the equivalences between the "schematic" and "imperative" computational models as follows.

Theorem.

(i) $PR(A) = \textbf{\textit{For}}(A)$

(ii) $PR^*(A) = \textbf{\textit{For}}^*(A)$

(iii) $\mu PR(A) = \textbf{\textit{While}}(A)$

(iv) $\mu PR^*(A) = \textbf{\textit{While}}^*(A)$,

in all cases, uniformly for $A \in \textbf{\textit{NStdAlg}}\,(\Sigma)$.

These results are all stated in [TZ00], and can be proved by the methods of [TZ88].

4.4 Universal Function Theorem for μPR*

The following is a uniform version of a theorem proved in [TZ00, §4.9] (using the equivalent formalism of **While*** programs):

Theorem. *For any Σ-function type $u \to s$, there is a $\mu PR^*(\Sigma)$ derivation $\upsilon : \mathsf{nat} \times u \to s$ which is universal for $\mu PR^*(\Sigma)$ derivations of type $u \to s$.*

In other words, we can enumerate all the μPR* derivations of type $u \to s$:

$$\alpha_0, \ \alpha_1, \ \alpha_2, \ \ldots$$

so that, putting

$$\varphi_i^A \ =_{df} \ \mathsf{f}_{\alpha_i}^A : A^u \ \to \ A_s$$

and

$$\boldsymbol{Univ}_{u \to s}^A \ = \ \mathsf{f}_\upsilon^A : \mathbb{N} \times A^u \to A_s$$

we have

$$\boldsymbol{Univ}_{u \to s}^A(i, a) \ = \ \varphi_i^A(a)$$

for all $A \in \boldsymbol{NStdAlg}\,(\Sigma)$ and $i = 0, 1, 2, \ldots$.

Remarks (Canonical forms of μPR* derivations). (1) From the construction of the universal $\mu PR^*(\Sigma)_{u \to s}$ derivation υ [TZ00, §4], it can be seen that υ uses the μ-operator exactly once.

(2) For any $\mu PR^*(\Sigma)_{u \to s}$ derivation α, the universal derivation $\upsilon : \mathsf{nat} \times u \to s$ provides an equivalent *canonical* or *normal form derivation* $\hat\alpha$, such that $\mathsf{f}_{\hat\alpha}^A = \mathsf{f}_\alpha^A$ for all N-standard Σ-algebras A. This canonical derivation is formed in a simple way from υ, essentially by substituting the Gödel number $\ulcorner\alpha\urcorner$ of α for the distinguished nat variable of υ, so that for all N-standard A,

$$\mathsf{f}_{\hat\alpha}^A \ = \ \varphi_{\ulcorner\alpha\urcorner}^A \ = \ \mathsf{f}_\alpha^A.$$

This is, in fact, a generalisation to $\boldsymbol{NStdAlg}\,(\Sigma)$ of Kleene's Normal Form Theorem for (essentially) μPR($\mathcal{N}$) [Kle52].

(3) From the constructions in (1) and (2) it follows that $\hat\alpha$ also uses the μ-operator exactly once, and in such a way that for any N-standard A,

$\mathsf{f}_{\hat\alpha}^A$ *is total if, and only if, this application of the μ-operator is total on A.*

5 Algebraic specifications for computable functions

We will consider functions f *computable* on a Σ-algebra, by PR and μPR* derivations, and show that they are algebraically *specifiable* by conditional equational, and conditional BU equational, theories.

We will also consider, in the context of *metric algebras* (*i.e.*, algebras with metrics such that the functions in the signature are continuous) a broader class of functions than

$\mu\mathrm{PR}^*$ computable, namely those functions *uniformly approximable* by $\mu\mathrm{PR}^*$ computable functions, and show that such functions are specifiable by *conditional equations and inequalities*, which are conditional formulae built up from inequalities $(t_1 < t_2)$ on the reals as well as equations $(t_1 = t_2)$ between terms of the same sort.

5.1 Algebraic specifications for PR computable functions

Let Σ be an N-standard signature. For each $\mathrm{PR}(\Sigma)$ derivation α, there is a finite set E_α of "specifying equations" for the function f, as well as the auxiliary functions $g = (g_1, \ldots, g_{k_\alpha})$, defined by α.

The set E_α consists of *equations* in an expanded signature $\Sigma_\alpha = \Sigma \cup \{\mathsf{g}_\alpha, \mathsf{f}_\alpha\}$, where $\mathsf{g}_\alpha \equiv \mathsf{g}_{\alpha,1}, \ldots, \mathsf{g}_{\alpha,k_\alpha}$. It is defined by course of values induction on the length of the derivation α, with cases (i)—(v) (§4.1) according to the last scheme in α. In fact, E_α is formed by adjoining, in each case, specifying equation(s) like those shown for that case in §4.1. These are simple (*i.e.*, not conditional) equations; for example, in the case (iv) *definition by cases*, there are two equations:

$$\mathsf{f}(\mathsf{true, x, y}) = \mathsf{x}$$
$$\mathsf{f}(\mathsf{false, x, y}) = \mathsf{y}$$

and in the case (v) *simultaneous primitive recursion*, there are $2m$ equations (where m is the degree of simultaneity):

$$\mathsf{f}_i(0, \mathsf{x}) = \mathsf{g}_i(\mathsf{x})$$
$$\mathsf{f}_i(\mathsf{z}+1, \mathsf{x}) = \mathsf{h}_i(\mathsf{z, x, f}_1(\mathsf{z, x}), \ldots, \mathsf{f}_m(\mathsf{z, x}))$$

for $i = 1, \ldots, m$.

Thus we have an effective map $\alpha \mapsto (\Sigma_\alpha, E_\alpha)$ from $\mathrm{PR}(\Sigma)$ derivations to (simple) equational specifications.

Now for each PR derivation α and N-standard Σ-algebra A, let f_α^A be the function on A computed by α, and let g_α^A be the corresponding auxiliary functions on A. Consider the operators

$$A \longmapsto (A, \mathsf{f}_\alpha^A) \tag{$*$}$$

and

$$A \longmapsto (A, \mathsf{g}_\alpha^A, \mathsf{f}_\alpha^A). \tag{$**$}$$

Recall the definition of uniform specification of an operator over a class of Σ-algebras (§3.1, Definitons 5 and 6).

Theorem 1 (Equational specification of PR functions). *For each $\mathrm{PR}(\Sigma)$ derivation α, the equational specification $(\Sigma_\alpha, E_\alpha)$ specifies the operator $(**)$ uniformly over $A \in$* ***NStdAlg*** *(Σ). Hence it specifies the operator $(*)$ uniformly over all N-standard Σ-algebras A, with hidden functions.*

Proof: By course of values induction on the length of α. $\quad\square$

In other words, the equations E_α specify not only f_α^A, but also the auxiliary functions g_α^A, uniformly over all N-standard Σ-algebras A.

Similarly with PR* computability: for a PR*(Σ) derivation α, let E_α be the set of specifying equations for the function f_α and the auxiliary functions g_α defined by α, in the signature $\Sigma_\alpha^* = \Sigma^* \cup \{\mathsf{g}_\alpha, \mathsf{f}_\alpha\}$.

Corollary. *For each PR*(Σ) derivation α, the equational specification $(\Sigma_\alpha^*, E_\alpha)$ specifies the operator $(*)$ uniformly over Σ, with hidden sorts and functions.*

5.2 Algebraic specifications for μPR* computable functions

We now consider μPR*(Σ) derivations α. For each such derivation there is again a finite set F_α of "specifying equations" for the function f defined by α and its auxiliary functions g. This set is constructed like E_α (§5.1), by structural induction on α. Now, however, F_α consists of *conditional BU equations* in a signature $\Sigma_\alpha^* = \Sigma^* \cup \{\mathsf{g}_\alpha, \mathsf{f}_\alpha\}$, because of the new case, *i.e.*, scheme (vi) for the μ-operator (§4.2), which results in the addition to F_α of the conditional BU equation

$$(F_\mu) \qquad \forall \mathsf{z} < \mathsf{y} \, (\mathsf{g}_0(\mathsf{x}, \mathsf{z}) = \mathsf{false}) \, \wedge \, (\mathsf{g}_0(\mathsf{x}, \mathsf{y}) = \mathsf{true}) \quad \longrightarrow \quad \mathsf{f}(\mathsf{x}) = \mathsf{y}.$$

Again we have an effective map $\alpha \mapsto (\Sigma_\alpha^*, F_\alpha)$ from μPR*(Σ) derivations to conditional BU equational specifications.

Now there are complications in the theory, since μPR* computable functions are, in general, partial. We intend to study specification theory for partial algebras systematically in a future paper. Here we limit ourselves to the case where the μPR* computable function is, in fact, total.

As before, for a μPR* derivation α and an N-standard Σ-algebra A, let f_α^A be the function on A defined by α, and let g_α^A be corresponding auxiliary functions on A^*. A further problem is that, even if f_α^A is total, the functions g_α^A might not be. We will now show that we can, without loss of generality, restrict attention to the case that the g_α^A are also total. We accomplish this by the use of the uniform derivations provided by the Universal Function Theorem for μPR* (§4.4), as we now explain.

Definition. A μPR* derivation α is *total on* A iff the auxiliary functions g_α^A, as well as f_α^A, are all total on A^*.

Totality Lemma. Given any μPR*(Σ) derivation $\alpha\colon u \to s$, we can effectively find a μPR*(Σ) derivation $\hat{\alpha}\colon u \to s$ such that for any N-standard Σ-algebra A,

  (i) $\mathsf{f}_{\hat{\alpha}}^A = \mathsf{f}_\alpha^A$;

  (ii) if f_α^A is total, then $\hat{\alpha}$ is total on A.

Proof: This follows from the Universal Function Theorem and the three remarks following it (§4.4). □

Now consider the operators $(*)$ (§5.1 above) and

$$A \;\longmapsto\; (A^*, \mathsf{g}_{\hat{\alpha}}^A, \mathsf{f}_{\hat{\alpha}}^A) \qquad\qquad (***)$$

where $\hat{\alpha}$ is constructed from α as in the totality lemma. Let $\Sigma_\alpha^* = \Sigma^* \cup \{\, \mathsf{g}_{\hat{\alpha}}, \mathsf{f}_{\hat{\alpha}} \,\}$. Recall the definition of the array specification $\mathsf{ArrAx}(\Sigma)$ in §3.2, and the definition of the conditional BU specification $F_{\hat{\alpha}}$ of $\mathsf{f}_{\hat{\alpha}}^A$ $(= \mathsf{f}_\alpha^A)$.

Theorem 2 (Conditional BU equational specification of μPR* functions).
For each $\mu PR^*(\Sigma)$ derivation α, let

$$F_\alpha^* \;=_{df}\; \mathsf{ArrAx}(\Sigma) \,+\, F_{\hat{\alpha}}$$

where $\hat{\alpha}$ is constructed from α as in the totality lemma. Then the conditional BU equational specification $(\Sigma_\alpha^*, F_\alpha^*)$ specifies the operator $(***)$ in the following sense: for any A on which f_α^A is total,

$$(A^*, \mathsf{g}_{\hat{\alpha}}^A, \mathsf{f}_{\hat{\alpha}}^A) \;\models\; F_\alpha^*.$$

Hence $(\Sigma_\alpha^*, F_\alpha^*)$ specifies the operator $(*)$ uniformly over all N-standard Σ-algebras A on which f_α^A is total, with hidden sorts and functions.

Proof: As with Theorem 1, by course of values induction on the length of α. $\square$

Note that the specification given in Theorem 2 is uniform over all N-standard Σ-algebras A on which α is total. In fact, there is a stronger form of uniformity for μPR* computability, following from the Universal Function Theorem for μPR*. (Actually, this is already implicit in the construction of the derivation $\hat{\alpha}$ in the totality lemma, which is really a normal form lemma for μPR* derivations.)

Theorem 3 (Universal conditional BU equational specification). *For each Σ-function type $u \to s$ we can effectively find a signature $\Sigma_{u,s}^*$ which expands Σ^* by function symbols, and a finite conditional BU equational specification $(\Sigma_{u,s}^*, F_{u,s}^U(\mathsf{z}))$ which is universal for specifications of total $\mu PR^*(\Sigma)$-computable functions of that type, in the following sense: it contains a distinguished number variable z such that for each $\mu PR^*(\Sigma)$ derivation $\alpha : u \to s$, and each N-standard Σ-algebra A, if f_α^A is total on A, then $(\Sigma_{u,s}^*, F_{u,s}^U(\bar{k}))$, where $k = \ulcorner \alpha \urcorner$, specifies f_α^A on A, with hidden sorts and functions.*

(Here $F_{u,s}^U(\bar{k})$ is the result of substituting the numeral $\bar{k}$ for z in $F_{u,s}^U(\mathsf{z})$.)

Next, by the BU Reduction Theorem (Theorem 2 in Section 3), we derive as a corollary to Theorem 3:

Theorem 4 (Universal conditional equational specification). *For each Σ-function type $u \to s$ we can effectively find a signature $\Sigma_{u,s}^{*\prime}$ which expands $\Sigma_{u,s}^*$ (of Theorem 3) by function symbols, and a finite conditional specification $(\Sigma_{u,s}^{*\prime}, E_{u,s}^U(\mathsf{z}))$ which is universal for specifications of total $\mu PR^*(\Sigma)$-computable functions of that type, in the following sense: it contains a distinguished number variable z such that for each*

$\mu PR^*(\Sigma)$ derivation $\alpha : u \to s$, and each N-standard Σ-algebra A, if f_α^A is total on A, then $(\Sigma_{u,s}^{*\prime}, E_{u,s}^U(\bar{k}))$, where $k = \ulcorner\alpha\urcorner$, specifies f_α^A on A, with hidden sorts and functions.

From the above uniformity theorems it follows trivially that for a given Σ-function type $u \to s$ there is a uniform bound to the lengths of conditional BU Σ^*-specifications, or conditional equational Σ^*-specifications respectively, for total μPR^*-computable functions on N-standard Σ-algebras.

6   Algebraic specifications for computably approximable functions

We have shown that

$$computability \implies algebraic\ specifiability$$

where (for example) if "computability" means μPR^* (or, equivalently, ***While****) computability, then "algebraic specifiability" means specifiability by conditional BU equations.

It is natural to ask in what sense a converse holds. We will see (below) that a full converse to the above cannot be expected, since algebraic specifiability is more powerful, in some sense, than μPR^*-computability. (In Section 7 we will investigate partial converses.) We show here in fact that, on metric algebras,

$$computable\ approximability \implies algebraic\ specifiability.$$

"Computable approximability", to be defined shortly, is a strong extension of the notion of computability; while "algebraic specifiability" will be (re-)defined so as to permit the order relation (as well as equality) between pairs of terms of sort real.

6.1   Metric algebras

We refer to [TZ99] and [TZ00, §6] for definitions of (total) *metric algebra* and related concepts. We review some definitions and results from these references. (Note that in these references the subject is discussed in the broader context of *partial algebras*.)

A *metric Σ-algebra (A, d), based on a Σ-algebra A*, is an algebra of the form

$$
\boxed{
\begin{array}{ll}
\textsf{algebra} & (A, d) \\
\textsf{import} & A \\
\textsf{carriers} & \mathbb{R} \\
\textsf{functions} & d_s : A_s^2 \to \mathbb{R} \quad (s \in \boldsymbol{Sort}(\Sigma)) \\
\textsf{end} &
\end{array}
}
$$

where d is a family $\langle d_s \mid s \in \boldsymbol{Sort}(\Sigma)\rangle$ of metrics d_s on the carriers A_s, where (in the case that A is standard or N-standard) d_{bool} and d_{nat} are the discrete metrics on $\mathbb{B}$ and $\mathbb{N}$ respectively, and such that the primitive functions on A are all continuous under these metrics.

We will often write 'd' for the metric d_s, and 'A' for the metric algebra (A, d).

Examples. (a) The metric algebra $\mathcal{R}^d$ on the reals ("d" for "distance") is defined by

```
algebra      R^d
import       R^N
functions    div_nat : ℝ × ℕ → ℝ,
             d_real : ℝ² → ℝ,
             d_nat : ℕ² → ℝ,
             d_bool : 𝔹² → ℝ
end
```

where $\mathcal{R}^N$ is the N-standard algebra of reals (§1.4, Example (b)), $\mathsf{div}_{\mathsf{nat}}$ is division of reals by naturals (where division by zero is defined as zero), the metric on $\mathbb{R}$ is the standard one, and the metrics on $\mathbb{N}$ and $\mathbb{B}$ are discrete.

Note that $\mathcal{R}^d$ does not contain the (total) boolean-valued functions $\mathsf{eq}_{\mathsf{real}}$ or $\mathsf{less}_{\mathsf{real}}$, since they are not continuous with respect to this metric.

(b) The interval metric algebra $\mathcal{I}^d$: Here the unit interval $I = [0,1]$ is included as a separate carrier of sort 'intvl', again with the usual metric. This is useful for studying real continuous functions with compact domain. (We could also choose $I = [-1,1]$, etc.) The algebra $\mathcal{I}^d$ is defined by

```
algebra      I^d
import       R^d
carriers     I
functions    ι_I : I → ℝ,
             d_intvl : I² → ℝ
end
```

where ι_I is the embedding of I into $\mathbb{R}$. Because of the importance of the metric algebra $\mathcal{I}^d$ as in our computation theory, let us review its construction. It contains $\mathbb{R}$ with its standard ring operations, $\mathbb{N}$ and $\mathbb{B}$ with their standard operations, functions for definition by cases on $\mathbb{R}$, $\mathbb{N}$ and $\mathbb{B}$, division of reals by naturals, the unit line interval $\mathbb{I}$ and its embedding in $\mathbb{R}$, and the standard metrics on all four carriers.

6.2 Definitions and theorems

Now let A be an N-standard metric Σ-algebra with metric d.

Definition 1 (μPR* computably approximable functions). A total function $f : A^u \to A_s$ on A is μPR* *computably approximable, uniformly on A*, if there is a total μPR* computable function

$$G : \mathbb{N} \times A^u \quad \to \quad A_s$$

and a total computable function $g : \mathbb{N} \to \mathbb{N}$ on A such that, putting $G_n =_{df} G(n, \cdot)$, the sequence G_n *approximates f uniformly* on A^u with modulus of approximation g, *i.e.*, for all n, k and all $x \in A^u$,

$$k \geq g(n) \implies d(G_k(x), f(x)) < 2^{-n}.$$

Definition 2 (Fast μPR* computably approximable functions). A total function $f : A^u \to A_s$ on A is *fast μPR^* computably approximable, uniformly on A*, if there is a total μPR* computable function $G : \mathbb{N} \times A^u \to A_s$ on A such that, putting $G_n =_{df} G(n, \cdot)$, the sequence G_n *approximates f uniformly fast* on A^u, i.e., for all n and all $x \in A^u$,

$$d(G_n(x), f(x)) \; < \; 2^{-n}. \tag{$*$}$$

Remark 1. It is easy to see that Definitions 1 and 2 are equivalent; for given a (computable) approximating sequence G_n with modulus of approximation g, we can effectively replace it by the fast (computable) approximating sequence $G'_n =_{df} G_n \circ g$. We will therefore usually tacitly assume w.l.o.g. that our approximating sequences are fast, and work with the (simpler) Definition 2.

Definition 3 (Fast μPR* approximating derivations). Let A be a metric Σ-algebra. A derivation $\gamma : \mathsf{nat} \times u \to s$ is an *approximating derivation for* a total function $f : A^u \to A_s$ if (i) the function $G : \mathbb{N} \times A^u \to A_s$ computed by γ on A is total on A; and (ii) G and f satisfy $(*)$ above.

Note that *at most one* function is μPR* approximable by a given derivation on any metric algebra.

Definition 4 (Conditional equation or inequality).
(a) A *conditional equation or inequality* is defined like a conditional equation, except that the atomic statements may be *either* equations $(t_1 = t_2)$ between terms of the same sort, *or* order $(t_1 < t_2)$ between terms of sort real.

(b) A *conditional BU equation or inequality* is defined like a conditional equation, except that the atomic statements may be *either* equations $(t_1 = t_2)$ or BU equations $(\forall \mathsf{z} < t \,[\, t_1 = t_2 \,])$ between terms of the same sort, or inequalities $(t_1 < t_2)$ between terms of sort real.

Remark 2. Here we are treating the order relation on the reals as a new atomic predicate of $\boldsymbol{Lang}(\Sigma^*)$ (like equality), *not* as a boolean-valued operation

$$\mathsf{less}_{\mathsf{real}} : \mathsf{real}^2 \to \mathsf{bool}.$$

This predicate (unlike such an operation) does not form part of the signature Σ. (See the analogous Remark concerning equality at the beginning of Section 2.)

Note also that $(*)$ is a conditional inequality (actually a simple inequality, without an antecedent).

Notation. We write μPR*-$\boldsymbol{Approx}(A)$ for the class of μPR* computably approximable functions on A, and μPR*-$\boldsymbol{Approx}(A)_{u \to s}$ for those of type $u \to s$.

In preparation for the next theorem, we note that a "Universal Function Theorem" holds for μPR*-$\boldsymbol{Approx}(A)$, in the following sense. For any Σ-function type $u \to s$, let

$$H^{u,s} =_{df} \boldsymbol{Univ}^A_{\mathsf{nat} \times u \to s} : \mathbb{N} \times \mathbb{N} \times A^u \to A_s$$

be the universal function for $\mu\mathrm{PR}^*(A)_{\mathsf{nat}\times u \to s}$ given by the Universal Function Theorem (§4.4). Then for each $f \in \mu\mathrm{PR}^*\text{-}\boldsymbol{Approx}(A)_{u\to s}$, there is a number k such that (writing $H^{u,s}_{k,n} = H^{u,s}(k,n,\cdot)$) the sequence of functions $H^{u,s}_{k,0}$, $H^{u,s}_{k,1}$, $H^{u,s}_{k,2}$, ... uniformly approximates f. The number k can be chosen as the Gödel number of an approximating derivation for f, *i.e.*, a derivation $\gamma\colon \mathsf{nat} \times u \to s$ of the function $H^{u,s}(k,\cdot)$. Combining this with Theorem 3 of Section 5, we obtain:

Theorem 1 (Universal conditional BU specification of $\mu\mathrm{PR}^*$ approximable functions). *For each Σ-function type $u \to s$ we can effectively find a signature $\Sigma^*_{u,s}$ which expands Σ^* by function symbols, and a finite conditional BU specification $(\Sigma^*_{u,s}, F^V_{u,s}(\mathsf{z}))$ consisting of conditional BU equations and inequalities, which is universal for specifications of $\mu PR^*(\Sigma)$-computably approximable functions of that type, in the following sense: it contains a distinguished number variable z such that for each $\mu PR^*(\Sigma)$ derivation $\gamma\colon \mathsf{nat} \times u \to s$ and each metric Σ-algebra A and total function $f\colon A^u \to A_s$, if γ is an approximating derivation for f on A, then $(\Sigma^*_{u,s}, F^V_{u,s}(\bar{k}))$, where $k = \ulcorner\gamma\urcorner$, specifies f on A, with hidden sorts and functions.*

Proof: Define
$$F^V_{u,s}(\mathsf{z}) \;=_{df}\; F^U_{u,s}(\mathsf{z}) + E_{\mathsf{invexp}} + E_*(\mathsf{z})$$

where $F^U_{u,s}(\mathsf{z})$ is the conditional BU equational specification constructed as in Theorem 3 in Section 5 for the universal function H for $\mu\mathrm{PR}^*$ computable functions of type $\mathsf{nat} \times u \to s$, E_{invexp} is the set of specifying equations for the computable real-valued function $\mathsf{invexp}(n) = 2^{-n}$, *i.e.*, its recursive definition:

$$\mathsf{invexp}(0) \;=\; 1, \qquad \mathsf{invexp}(\mathsf{S}n) \;=\; \mathsf{div}_{\mathsf{nat}}(\mathsf{invexp}(n), 2),$$

and $E_*(\mathsf{z})$ is the inequality $(*)$ above — or rather, its formal version

$$\mathsf{d}(\mathsf{H}(\mathsf{z}, \mathsf{n}, \mathsf{x}), \mathsf{f}(\mathsf{x})) \;<\; \mathsf{invexp}(\mathsf{n}). \qquad\qquad (**)$$

(Note that every $\mu\mathrm{PR}^*$ function $G\colon \mathbb{N}\times A^u \to A_s$ is obtainable from H by substituting the Gödel number of its derivation for the first argument z of H.) Let $\Sigma^*_{u,s}$ be the signature formed by expanding Σ^* by symbols for H and invexp, as well as the auxiliary functions used in their computations. Then for any $\mu\mathrm{PR}^*$ derivation $\gamma\colon \mathsf{nat} \times u \to s$, metric Σ-algebra A and function $f\colon \mathbb{N} \times A^u \to A_s$, if γ is an approximating derivation for f then $(\Sigma^*_{u,s}, F^V_{u,s}(\bar{k}))$ (where $k = \ulcorner\gamma\urcorner$) is a conditional BU specification of f on A, with hidden sorts and functions, consisting of conditional BU equations and inequalities   □

Now, by adapting the BU Reduction Theorem (§3.3) to specifications with inequalities, we derive as a corollary to Theorem 1:

Theorem 2 (Universal conditional specification of μPR* approximable functions). *For each Σ-function type $u \to s$ we can effectively find a signature $\Sigma_{u,s}^{*\prime}$ which expands $\Sigma_{u,s}^{*}$ (of Theorem 1) by function symbols, and a finite conditional specification $(\Sigma_{u,s}^{*\prime}, E_{u,s}^{V}(\mathsf{z}))$, consisting of conditional equations and inequalities, which is universal for specifications of approximably $\mu PR^{*}(\Sigma)$-computable functions of that type, in the following sense: it contains a distinguished number variable z such that for each $\mu PR^{*}(\Sigma)$ derivation $\gamma \colon \mathsf{nat} \times u \to s$ and each metric Σ-algebra A and total function $f \colon A^{u} \to A_{s}$, if γ is an approximating derivation for f on A, then $(\Sigma_{u,s}^{*\prime}, E_{u,s}^{V}(\bar{k}))$, where $k = \ulcorner \gamma \urcorner$, specifies f on A, with hidden sorts and functions.*

Remark 3 (Replacing the order predicate by a boolean-valued operation). The order relation in the above specification is used in one place only: in the (conditional) relation $(*)$ (or $(**)$). In fact (Remark 2 above notwithstanding) $(*)$ *could* be interpreted as a conditional *equation* (so that f is *conditionally equationally definable* with hidden sorts and functions) by interpreting '$t_1 < t_2$' as '$\mathsf{less_{real}}(t_1, t_2) = \mathsf{true}$', where the boolean-valued operator

$$\mathsf{less_{real}} : \mathsf{real}^2 \to \mathsf{bool}$$

is included in the signature of the metric algebra over $\mathbb{R}$. The problem here is that (as discussed in [TZ99]) whereas all functions in the signature of metric algebras (and hence all functions computable over these) are *continuous*, the (total) function $\mathsf{less_{real}}$ is discontinuous. The only way to restore continuity is to consider a *partial* continuous $\mathsf{less_{real}}$ operator, which leads to a study of *topological partial algebras*. This can be done, and the whole of the present theory could be re-cast in such a context, but that would take us too far afield from the present study.

Let us apply Theorem 2 to the classical notion of *Gzegorczyk-Lacombe (GL) computability* on the unit interval $I = [0, 1]$. This includes all the well-know functions of real analysis (sin, exp, log, etc.) restricted to I.

Notation. We write $\mathrm{GL}_m^{\mathrm{T}}(I)$ for the class of GL-computable total functions $f \colon I^m \to \mathbb{R}$.

Many concrete models of computation on $\mathbb{I}$ are equivalent to this class [PER89, Wei00]. It has been shown that (recall the definition of $\mathcal{I}^d$ in Example (b)):

$$\mathrm{GL}_m^{\mathrm{T}}(I) \ = \ \mu\mathrm{PR}^{*}\text{-}\boldsymbol{Approx}(\mathcal{I}^d)_{\mathsf{intvl}^m \to \mathsf{real}}.$$

(For details, see [TZ99, §9], [TZ00, §5.9].) Hence, again, a kind of "Universal Function Theorem" holds for $\mathrm{GL}_m^{\mathrm{T}}(I)$, in the following sense. For $m = 1, 2, \ldots,$ let

$$H^m \ =_{df} \ \boldsymbol{Univ}_{\mathsf{nat} \times \mathsf{intvl}^m \to \mathsf{real}}^{\mathcal{I}^d} : \ \mathbb{N} \times \mathbb{N} \times I^m \ \to \ \mathbb{R}$$

be the universal function for $\mu\mathrm{PR}^{*}(\mathcal{I}^d)_{\mathsf{nat} \times \mathsf{intvl}^m \to \mathsf{real}}$ given by the Universal Function Theorem (§4.4). Then for each $f \in \mathrm{GL}_m^{\mathrm{T}}(I)$, there is a number k, effectively obtainable from the GL-code for f, such that (writing $H_{k,n}^m = H^m(k, n, \cdot)$) the sequence of functions $H_{k,0}^m, H_{k,1}^m, H_{k,2}^m, \ldots$ uniformly approximates f on I.

So by Theorem 2 applied to $\mathcal{I}^d$:

Theorem 3 (Universal specification of GL computable functions). *For each $m > 0$ there is a signature Σ_m^* which is an expansion of $\Sigma(\mathcal{I}^d)^*$ by finitely many function symbols, and a finite conditional specification $(\Sigma_m^*, E_m^W(\mathbf{z}))$ consisting of conditional equations and inequalities, which is universal for specifications of $\mathrm{GL}_m^{\mathrm{T}}(I)$, in the following sense: it contains a distinguished natural number variable $\mathbf{z}$ such that each function $f \in \mathrm{GL}_m^{\mathrm{T}}(I)$ is specified (with hidden sorts and functions) by a suitable substitution instance $(\Sigma_m^*, E_m^W(\bar{k}))$, where k can be found effectively from a GL-code for f.*

Remark 4 (Description of the signature Σ_m^* of Theorem 3). The signature Σ_m^* is an expansion of $\Sigma(\mathcal{I}^d)$ (for a description of which see the remark at the end of §6.1) by the following sorts and functions:

(i) the sorts and functions of the array structure over $\mathcal{I}^d$ (§1.6);

(ii) the $\mu\mathrm{PR}^*$ "universal function" F^m for $\mathrm{GL}_m^{\mathrm{T}}(I)$ (as described in the above discussion) together with the auxiliary functions in its derivation;

(iii) the function 2^{-n}, used for assertions about computable approximations, as explained in the proof of Theorem 1;

(iv) the characteristic function for BU quantification, as described in the proof of the BU elimination theorem (§3.3).

Note that there is only one function of type (iv) in Σ_m^*, namely that obtained by eliminating the conditional BU equation F_μ (§5.2) specifying the (single!) μ-operator occurrence in the $\mu\mathrm{PR}^*$ derivation for H^m in (ii) (see Remark 1 in §4.4). (There are no conditional BU equality axioms for arrays (§3.2) to eliminate here, since real and intvl are not equality sorts.)

6.3 Illustration: Specification of dynamical system

We illustrate the connection between algebraic specification methods and models of physical systems.

A *deterministic dynamical system* with finite dimensional state space $S \subseteq \mathbb{R}^n$ and time $T \subseteq \mathbb{R}$ is represented in a model by a function

$$\phi: \ T \times S \ \to \ S$$

where for $t \in T$, $s \in S$, $\phi(t, s)$ is the state of the system at time t with initial state s. For example, the state of a particle in motion is represented by position and velocity. Thus, for a system of n particles in 3-dimensional space, the state space has $6n$ dimensions.

In practice, the model is specified by ordinary differential equations (ODEs) whose complete solution is ϕ. Specifically, in the modern qualitative theory of ODEs [Arn73], ϕ is differentiable, and the function $\phi_t : S \to S$ defined by

$$\phi_t(s) \ = \ \phi(t, s) \quad \text{for} \ \ t \in T, s \in S,$$

is a 1-parameter group of diffeomorphisms of S; the action of this group on S is called the *flow* on the *phase space* S. This flow can be specified by a vector field on S.

In modelling a physical system, one aim is to compute values of the function ϕ on some time interval and subspace of the space of initial conditions. Many methods exist to derive algorithms for ϕ from the equations that define it. Indeed, various fields of applied mathematics exist in order to design such equations, and the field of numerical analysis exists to design such solution methods.

Conversely, we suppose that ϕ can be simulated on a digital computer, *i.e.*, ϕ is a classically computable (*e.g.*, GL-computable) function. *Assume also that the state space S is the unit n-cube I^n, and the time dimension T is the unit interval I.* Thus

$$\phi\colon I \times I^n \ \to \ I.$$

We can now apply Theorem 3 to show that the dynamical system has a finite algebraic specification.

Theorem 4 (Universal specification of computable dynamical systems). *For each $n > 0$ there is a signature $\tilde{\Sigma}_n^*$ which extends $\Sigma(\mathcal{I}^d)^*$ by function symbols, and a finite conditional specification $(\tilde{\Sigma}_n^*, \tilde{E}_n^W(\mathbf{z}))$ consisting of conditional equations and inequalities, which is universal for all classically computable dynamical systems on the unit n-cube I^n over the unit time interval I.*

Note that $\tilde{\Sigma}_n^*$ is essentially the signature Σ_m^* of Theorem 3, with $n = m + 1$.

We have shown above how powerful algebraic specifications are, even for topological data types. More research needs to be done to determine the extent of its power, especially on metric algebras. Here topological notions such as continuity can play a part, as we see from the following example.

6.4   Example: Specification of function assuming continuity

Consider the two equations for a (total) function f on the real line:

$$f(x + y) \ = \ f(x) \times f(y), \qquad\qquad f(1) \ = \ c \qquad\qquad\qquad (*)$$

for some constant $c > 0$. If we assume that f is continuous, even at one point, then it is easy to see that these equations are satisfied uniquely by the function

$$f(x) \ = \ c^x.$$

However, in the absence of any such continuity assumption, it can be shown that (for any $c > 0$) there are $2^{2^{\aleph_0}}$ non-constructive solutions to $(*)$. Here "non-constructive" means both that these solutions are non-computable, and that their existence is proved by non-constructive means, using Zorn's Lemma to show the existence of a Hamel basis on $\mathbb{R}$, *i.e.*, a maximal linear independent subset of $\mathbb{R}$ over $\mathbb{Q}$.

Note that any solution f of $(*)$ is a homomorphism from the additive group of reals to the multiplicative group of positive reals.

This example suggests the following

Question. *On metric algebras, does conditional equational specifiability, together with a topological condition such as continuity, imply computability?*

Specifically, is there a continuous function on I which is definable by equations but not approximably computable?

Note, in this connection, that there are other "equational specifications" for the exponential function e^x:

(1) the differential equation $f'(x) = f(x)$ with initial condition $f(0) = 1$;

(2) from the polynomial approximations given by the partial sums of the Maclaurin expansion $\sum_{i=0}^{\infty} x^i/i!$, a specification consisting of conditional equations and inequalities can be derived by the methods of this section for approximating computations;

(3) similarly, from the polynomial approximations, $(1 + x/n)^n$, a specification consisting of conditional equations and inequalities can be derived.

Note that in the first of these specifications, differentiability of f is (of course) implicitly assumed, and uniqueness of the solution follows by the Lipschitz condition; however no assumptions of continuity are needed in (2) or (3).

The above question points to an open field of research. The investigation of computable solutions of recursive equations in [GH00] would be relevant here.

7 Initial algebra specifications with conditional equations and conditional BU equations

In this section we will consider theories T, which we assume to be formalised in logical formalisms $\mathcal{F}$ of the kind described in Section 2; for example, $\mathcal{F} = \mathsf{CondBUEq}(\Sigma)$.

7.1 Pre-initial and initial models

In this subsection (only), we make no assumptions concerning the (N-)standardness of signatures or algebras. Let Σ be a signature and let $\mathbb{K}$ be a Σ-adt.

A formalism $\mathcal{F}$ is said to be *valid for* $\mathbb{K}$ if the axioms and inference rules of $\mathcal{F}$ hold for all algebras in $\mathbb{K}$. Note, for example, that $\mathsf{CondBUEq}(\Sigma)$ is valid for $\boldsymbol{NStdAlg}\,(\Sigma)$, but not, in general, for $\boldsymbol{Alg}\,(\Sigma)$.

A Σ-algebra A is *pre-initial for* $\mathbb{K}$ if there is a unique Σ-homomorphism from A to every algebra in $\mathbb{K}$; *pre*-initial in that it might not itself belong to $\mathbb{K}$. (The notion of Σ-*homomorphism* between Σ-algebras is defined as usual [MT92].)

Note that the *closed term algebra* $\boldsymbol{T}(\Sigma)$ is pre-initial for $\mathbb{K}$.

An *initial algebra of* $\mathbb{K}$ is a pre-initial algebra which belongs to $\mathbb{K}$. As is easily seen, any two initial algebras of $\mathbb{K}$ must be Σ-isomorphic. We denote any initial algebra of $\mathbb{K}$ by $\boldsymbol{Init}(\Sigma, \mathbb{K})$.

We will be interested in the case that

$$\mathbb{K} \;=\; \boldsymbol{Alg}(\varSigma, T),$$

the class of models of a first-order $\varSigma$-theory T, where T may have certain syntactic restrictions. We will assume:

- in this subsection that T is a conditional equational theory;
- in §7.2 likewise, but restrict attention to N-standard models of T;
- in §7.3 that T is a conditional BU equational theory (again with N-standard models);
- and in §7.4 that it is a conditional SU equational theory (ditto).

(Recall the formal systems defined in Section 2.) Finally in §7.5 we will show how conditional BU equational initial algebra specifications can be "reduced" to conditional equational initial algebra specifications.

Let T be a $\varSigma$-theory. We write $\boldsymbol{Init}(\varSigma, T)$ for the initial algebra $\boldsymbol{Init}\big(\varSigma, \boldsymbol{Alg}(\varSigma, T)\big)$ (if it exists), and call it the *initial model of T*.

Consider the *closed term algebra* $\boldsymbol{T}(\varSigma, T, \mathcal{F})$ formed from $\boldsymbol{T}(\varSigma)$ by identifying closed terms provably equal from T, in some formalism $\mathcal{F}$, *i.e.*,

$$\boldsymbol{T}(\varSigma, T, \mathcal{F}) \;=_{df}\; \boldsymbol{T}(\varSigma)/\!\approx_{T,\mathcal{F}}$$

where

$$t_1 \approx_{T,\mathcal{F}} t_2 \quad \Longleftrightarrow_{df} \quad t_1 = t_2 \text{ is provable from } T \text{ in } \mathcal{F}.$$

Lemma. If $\mathcal{F}$ is valid for $\boldsymbol{Alg}(\varSigma, T)$, then $\boldsymbol{T}(\varSigma, T, \mathcal{F})$ is pre-initial for $\boldsymbol{Alg}(\varSigma, T)$.

We will investigate whether $\boldsymbol{T}(\varSigma, T, \mathcal{F})$ is, further, *initial* for $\boldsymbol{Alg}(\varSigma, T)$, *i.e.*, whether

$$\boldsymbol{T}(\varSigma, T, \mathcal{F}) \;=\; \boldsymbol{Init}(\varSigma, T).$$

Initiality Lemma. *Suppose $\mathcal{F}$ is valid for $\boldsymbol{Alg}(\varSigma, T)$. If $\boldsymbol{T}(\varSigma, T, \mathcal{F}) \in \boldsymbol{Alg}(\varSigma, T)$, then it is ($\varSigma$-isomorphic to) $\boldsymbol{Init}(\varSigma, T)$.*

Definitions. Let A be a $\varSigma$-algebra.

(1) A has an *initial algebra specification* $(\varSigma, T)$ if $A \cong \boldsymbol{Init}(\varSigma, T)$.

(2) A has an *initial algebra specification with hidden sorts and/or functions* $(\varSigma', T')$ if $\varSigma'$ is an expansion of $\varSigma$ by sorts and/or functions, T' is a $\varSigma'$-theory and

$$A \;\cong\; \boldsymbol{Init}\big(\varSigma, \, \boldsymbol{Alg}(\varSigma', T')\,|_{\varSigma}\big).$$

Theorem 1 [Mal73]. *Let E be a conditional equational theory over Σ. Let $I =_{df}$ $\boldsymbol{T}(\Sigma, E, \mathsf{CondEq}(\Sigma))$. Then I is an initial model of E. Furthermore, if t_1, t_2 are two closed Σ-terms of the same sort, then the following are equivalent:*

(i) t_1 and t_2 have the same value in I,

(ii) t_1 and t_2 have the same value in all models of E,

(iii) $t_1 = t_2$ is provable from E in $\mathsf{CondEq}(\Sigma)$,

(iv) $t_1 = t_2$ is provable from E in $\mathsf{FOL}(\Sigma)$.

Proof: The main thing here is to show that $I \models E$, from which $(ii) \Rightarrow (i)$ will follow. Since I is a (closed) term model, it is sufficient to show that I satisfies all *closed substitution instances* of the axioms of E. So consider any closed instance $P_1 \wedge \ldots \wedge P_n \to P$ of an axiom of E, where P_i and P are closed equations. Note that the corresponding sequent

$$P_1, \ldots, P_n \longmapsto P \tag{$*$}$$

is derivable from E in $\mathsf{CondEq}(\Sigma)$, by the substitution rule. Suppose $I \models P_i$ for $i = 1, \ldots, n$. Then, by the definition of I, P_i is provable from E in $\mathsf{CondEq}(\Sigma)$. But then P is also provable, by repeated (atomic) cuts of the sequent $(*)$ with the sequents $\longmapsto P_i$, and so $I \models P$.

Hence $I \models E$. It follows, by the Initiality Lemma, that I is an *initial* model of E. Hence also $(ii) \Rightarrow (i)$. The further implications $(i) \Rightarrow (iii) \Rightarrow (iv) \Rightarrow (ii)$ are all trivial. $\square$

Remark (Completeness and conservativity). Mal'cev's Theorem [Mal73], in the form given above, can be viewed as expressing both (a) *completeness* of $\mathsf{CondEq}(\Sigma)$, given by the implication $(ii) \implies (iii)$, and (b) *conservativity* of first order logic with equality over $\mathsf{CondEq}(\Sigma)$, given by the implication $(iv) \implies (iii)$. (*Cf.* conservativity lemma (1) and the remark in §2.6.)

Necessary and sufficient conditions for the existence of initial models of theories are given in [MM84].

7.2 Initial N-standard models

Assume, from now on, that Σ is N-standard, and that $\mathbb{K}$ consists of *N-standard* Σ-algebras; for example, $\mathbb{K} = \boldsymbol{NStdAlg}(\Sigma, T)$, for some Σ-theory T. Then $\boldsymbol{T}(\Sigma, T, \mathcal{F})$, although it is pre-initial for $\mathbb{K}$, might fail to be initial for $\mathbb{K}$ for two reasons: it might not satisfy T, and it might not even be N-standard! (We return to the second point below.)

An *initial N-standard model of* T is an initial algebra of $\boldsymbol{NStdAlg}(\Sigma, T)$. Any two initial N-standard models of T are Σ-isomorphic. We denote any such model by

$$\boldsymbol{InitNStdAlg}(\Sigma, T) \ =_{df} \ \boldsymbol{Init}(\Sigma, \boldsymbol{NStdAlg}(\Sigma, T)).$$

N-Standard Initiality Lemma. *Suppose $\mathcal{F}$ is valid for $\boldsymbol{NStdAlg}(\Sigma, T)$. If $\boldsymbol{T}(\Sigma, T, \mathcal{F}) \in \boldsymbol{NStdAlg}(\Sigma, T)$ then it is (Σ-isomorphic to) $\boldsymbol{InitNStdAlg}(\Sigma, T)$.*

Definitions. Let A be an N-standard Σ-algebra.

(1) A has an *initial N-standard algebra specification* (Σ, T) if $A \cong \boldsymbol{InitNStdAlg}(\Sigma, T)$.

(2) A has an *initial N-standard algebra specification with hidden sorts and/or functions* (Σ', T') if Σ' is an expansion of Σ by sorts and/or functions, T' is a Σ'-theory and

$$A \cong \boldsymbol{Init}\big(\Sigma, \, \boldsymbol{NStdAlg}(\Sigma', T') \!\mid_{\Sigma}\big).$$

Note that $\boldsymbol{InitNStdAlg}(\Sigma, T)$ (if it exists) might not be an initial model of T, *i.e.*, T might have another, non-N-standard, initial model, as the following example demonstrates.

Example (Initial N-standard model of a theory which is not an initial model of that theory). Let Σ contain (in addition to the standard operations on nat and bool) a constant $\mathsf{u} : \mathsf{bool}$, and let T contain the single axiom '$\mathsf{u} \neq \mathsf{true}$'. Then the term algebra $\boldsymbol{T}(\Sigma)$ trivially satisfies T, and is hence (by the Initiality Lemma of §7.1) an *initial* model of T. It is *not N-standard*, since it has a 3-element carrier of sort bool, with distinct denotations of true, false and u. There is, however, also an *initial N-standard* model of T with an N-standard (2-element) carrier of sort bool, formed by identifying u and false.

Now $\boldsymbol{T}(\Sigma, T, \mathcal{F})$ may fail to be N-standard for two reasons: that T proves "too little" or "too much", roughly speaking. The first reason is connected with non-N-standard interpretations of the sorts nat and bool. Thus, there may be a function symbol f in Σ with range sort nat, without corresponding axioms in T capable of "reducing" $f(t)$, for some closed term t, to a numeral. Similarly (as in the above example), not all closed boolean terms (*i.e.*, terms of sort bool) may be (provably in T) equal to true or false. (In the terminology of [GH78] the specification (Σ, T) is not "sufficiently complete".) The second reason is that T may be *inconsistent*, in the sense that it proves 'true = false' (or, equivalently in a suitable weak background theory, '$0 = 1$'). This motivates the following definitions. Note that we must (to begin with) speak of provability relative to some formal system $\mathcal{F}$, which will typically be one of the system $\mathsf{CondEq}(\Sigma)$ or $\mathsf{CondBUEq}(\Sigma)$ of Section 2.

Definition 3. T is *consistent* in $\mathcal{F}$ if the equation 'true = false' is not provable in $\mathcal{F}$ from T.

Definition 4. T *determines* nat in $\mathcal{F}$ if every closed term of sort nat is, provably in $\mathcal{F}$ from T, equal to a numeral; and T *determines* bool in $\mathcal{F}$ if every closed term of sort bool is, provably in $\mathcal{F}$ from T, equal to true or false.

Definition 5 (N-standardness axioms).

(a) $\mathsf{NStdAx}(\Sigma)$ is the following set of conditional equations:

$$
\begin{aligned}
&\mathsf{and}(\mathsf{true},\mathsf{true}) = \mathsf{true}, \qquad \mathsf{and}(\mathsf{true},\mathsf{false}) = \mathsf{and}(\mathsf{false},\mathsf{true}) = \mathsf{and}(\mathsf{false},\mathsf{false}) = \mathsf{false}, \\
&\qquad \mathsf{or}(\mathsf{false},\mathsf{false}) = \mathsf{false}, \qquad \mathsf{or}(\mathsf{true},\mathsf{true}) = \mathsf{or}(\mathsf{true},\mathsf{false}) = \mathsf{or}(\mathsf{false},\mathsf{true}) = \mathsf{true}, \\
&\qquad\qquad\qquad \mathsf{not}(\mathsf{true}) = \mathsf{false}, \qquad \mathsf{not}(\mathsf{false}) = \mathsf{true}, \\
&\qquad\qquad \mathsf{if}_s(\mathsf{true},\, \mathrm{x}_1^s,\mathrm{x}_2^s) \ = \ \mathrm{x}_1^s, \qquad \mathsf{if}_s(\mathsf{false},\, \mathrm{x}_1^s,\mathrm{x}_2^s) \ = \ \mathrm{x}_2^s, \\
&\qquad\qquad \mathsf{eq}_{\mathsf{nat}}(0,0) = \mathsf{true}, \qquad \mathsf{eq}_{\mathsf{nat}}(\mathsf{S}z,0) = \mathsf{eq}_{\mathsf{nat}}(0,\mathsf{S}z) = \mathsf{false}, \\
&\qquad\qquad\qquad\qquad \mathsf{eq}_{\mathsf{nat}}(\mathsf{S}z_1,\mathsf{S}z_2) = \mathsf{eq}_{\mathsf{nat}}(z_1,z_2), \\
&\qquad\qquad\qquad \mathsf{less}_{\mathsf{nat}}(0,\mathsf{S}z) = \mathsf{true}, \qquad \mathsf{less}_{\mathsf{nat}}(z,0) = \mathsf{false}, \\
&\qquad\qquad\qquad\qquad \mathsf{less}_{\mathsf{nat}}(\mathsf{S}z_1,\mathsf{S}z_2) = \mathsf{less}_{\mathsf{nat}}(z_1,z_2), \\
&\qquad\qquad\qquad\qquad\qquad \mathsf{eq}_s(\mathrm{x}^s,\mathrm{x}^s) = \mathsf{true}, \\
&\qquad\qquad\qquad\qquad \mathsf{eq}_s(\mathrm{x}_1^s,\mathrm{x}_2^s) = \mathsf{true} \ \rightarrow \ t_1^s = t_2^s.
\end{aligned}
$$

where, in the axioms for if_s, s ranges over all Σ-sorts other than bool; and in the axioms for eq_s, s ranges over all Σ-*equality sorts* other than nat,

(b) $\mathsf{NStdAx}_0(\Sigma)$ is the set of all *closed Σ-substitution instances* of $\mathsf{NStdAx}(\Sigma)$.

   Note that $\mathsf{NStdAx}(\Sigma) + \mathsf{Ind}(\Sigma)$ holds in any N-standard Σ-algebra.

   We use the terminology: *T proves $\mathsf{NStdAx}_0(\Sigma)$ in $\mathcal{F}$* to mean: $\mathsf{NStdAx}_0(\Sigma)$ is derivable from T in $\mathcal{F}$.

   We now state some lemmas which give sufficient conditions for a term model $\boldsymbol{T}(\Sigma, T, \mathcal{F})$ to be N-standard.

Lemma 1 (N-standardness lemma). *Suppose that in $\mathcal{F}$*

   (i) T is consistent,

   (ii) T determines nat and bool, and

(iii) T proves $\mathsf{NStdAx}_0(\Sigma)$.

Then $\boldsymbol{T}(\Sigma, T, \mathcal{F})$ is N-standard.

Lemma 2. *If Σ is strictly N-standard then $\mathsf{NStdAx}_0(\Sigma)$ determines nat and bool in $\mathsf{CondEq}(\Sigma)$.*

Proof: By structural induction on all closed Σ-terms of sort nat and bool (simultaneously). $\square$

   The following is an immediate consequence of Lemmas 1 and 2.

Lemma 3 (Strict N-standardness lemma). *Suppose Σ is strictly N-standard, $\mathcal{F}$ is at least as strong as* $\mathsf{CondEq}(\Sigma)$, *and in* $\mathcal{F}$

(i) *T is consistent, and*

(ii) *T proves $\mathsf{NStdAx}_0(\Sigma)$ (or $\mathsf{NStdAx}(\Sigma)$).*

Then $\boldsymbol{T}(\Sigma, T, \mathcal{F})$ is N-standard.

7.3 Conditional equational theories

We now give the analogue of Mal'cev's Theorem (§7.1) for N-standard models of conditional equational theories.

Theorem 2. *Let E be a conditional equational theory over Σ. Suppose that in* $\mathsf{CondEq}(\Sigma)$, *E is consistent, determines nat and bool, and proves $\mathsf{NStdAx}_0(\Sigma)$. Then $I =_{df} \boldsymbol{T}(\Sigma, E, \mathsf{CondEq}(\Sigma))$ is an initial N-standard model of E. Furthermore, if t_1, t_2 are two closed Σ-terms of the same sort, then the following are equivalent:*

(i) *t_1 and t_2 have the same value in I,*

(ii) *t_1 and t_2 have the same value in all N-standard models of E,*

(iii) *$t_1 = t_2$ is provable from E in $\mathsf{CondEq}(\Sigma)$,*

(iv) *$t_1 = t_2$ is provable from E in $\mathsf{FOL}(\Sigma) + \mathsf{Ind}(\Sigma)$.*

Proof: By the N-standardness Lemma (§7.2), I is an N-standard algebra. As in Theorem 1, the main thing is to show that $I \models E$. This is done exactly as in the proof of Theorem 1. It follows, by the N-standard Initiality Lemma (§7.2), that I is an *initial* N-standard model of E. The rest of the proof is similar to that for Theorem 1. Note for the implication $(iv) \Rightarrow (ii)$, we use the fact that the rule $\mathsf{Ind}(\Sigma)$ is valid for N-standard Σ-algebras. □

Remarks. (1) By Lemma 2 in §7.2, the assumption in the theorem that E determines nat and bool can be replaced by the assumption that Σ is strictly N-standard.

(2) (*Completeness and conservativity.*) Here again, the implication $(ii) \Rightarrow (iii)$ can be construed as a completeness theorem, and $(iv) \Rightarrow (iii)$ as a conservativity theorem. (See the Remark in §2.6 and the Remark following Theorem 1.)

(3) (*The N-standardness axioms.*) We have "incorporated" the N-standardness axioms $\mathsf{NStdAx}_0(\Sigma)$ in the theory E, so to speak, by assuming that E proves them. Another feasible approach would be to incorporate these axioms in the logics CondEq, $\mathsf{CondBUEq}$ and FOL, by adding them as axioms (as we did with the boundedness axioms BddAx in $\mathsf{CondBUEq}$). This would entail some minor re-wording of the theorems.

We turn our attention to theories with syntactic structure more complicated than conditional equations.

7.4 Conditional BU equational theories

We give the analogue of Mal'cev's Theorem for N-standard models of BU conditional equational theories.

Theorem 3. *Let F be a conditional BU equational theory over Σ. Suppose that in $\mathsf{CondBUEq}(\Sigma)$, F is consistent, determines nat and bool and proves $\mathsf{NStdAx_0}(\Sigma)$. Then $I =_{df} \boldsymbol{T}(\Sigma, F, \mathsf{CondBUEq}(\Sigma))$ is an initial N-standard model of F. Furthermore, if t_1, t_2 are two closed Σ-terms of the same sort, then the following are equivalent:*

(i) t_1 and t_2 have the same value in I,

(ii) t_1 and t_2 have the same value in all N-standard models of F,

(iii) $t_1 = t_2$ is provable from F in $\mathsf{CondBUEq}(\Sigma)$,

(iv) $t_1 = t_2$ is provable from F in $\mathsf{FOL}(\Sigma) + \mathsf{Ind}(\Sigma)$.

Proof: By the N-standardness Lemma, I is N-standard. As in Theorems 1 and 2, the main thing is to show that $I \models F$. Again, since I is a term model, it is sufficient to show that I satisfies the set of closed substitution instances of F. First note that, by definition, I satisfies precisely all closed equations provable from F in CondBUEq, *i.e.*, for any *closed equation P*:

$$I \models P \iff F \vdash P \qquad\qquad (*)$$

where '$\vdash$' here means provability in $\mathsf{CondBUEq}$. Further, by use of the *boundedness axioms* BddAx of $\mathsf{CondBUEq}$ (§2.3), the same holds for any *closed BU equation Q*:

$$I \models Q \iff F \vdash Q. \qquad\qquad (**)$$

For suppose $Q \equiv \forall \mathsf{z} < t P(\mathsf{z})$, where $P(\mathsf{z})$ is an equation. Since I is N-standard,

$$I \models t = \bar{n} \qquad\qquad (***)$$

for some (unique) n. Then

$$
\begin{aligned}
I \models \forall \mathsf{z} < t P(\mathsf{z}) &\iff \text{ for all } k < n,\ I \models P(\bar{k}) \\
&\iff \text{ for all } k < n,\ F \vdash P(\bar{k}) \qquad \text{by } (*) \\
&\iff F \vdash \forall \mathsf{z} < t P(\mathsf{z}) \qquad\qquad \text{by } \mathsf{BddAx} \text{ and } (***).
\end{aligned}
$$

Now consider any closed instance $f \equiv Q_1 \wedge \ldots \wedge Q_m \to Q$ of an axiom of F (where Q_i and Q are closed SU equations). Suppose $I \models Q_i$ for $i = 1, \ldots, m$. Then by $(**)$ Q_i is provable from F in $\mathsf{CondBUEq}$. But then so is Q, by repeated cuts of the sequent $Q_1, \ldots, Q_m \longmapsto Q$ corresponding to f with the sequents $\longmapsto Q_i$, and so $I \models Q$. $\square$

Remarks. (1) As before, the assumption in the theorem that F determines nat and bool can be replaced by the assumption that Σ is strictly N-standard.

(2) (*Completeness and conservativity.*) Again, the implication $(ii) \Rightarrow (iii)$ can be construed as a completeness theorem, and $(ii) \Rightarrow (iii)$ as a conservativity theorem.

7.5   Conditional SU equational theories

Now we turn to the infinitary conditional SU equational logic (§2.4). Although it will not be used further in the paper, it is interesting in its own right.

Remember that the infinitary ω-rule $\forall_\omega R$ obviates the need for an induction rule.

Theorem 4. *Let G be a conditional SU equational theory over Σ. Suppose that in $\mathsf{CondSUEq}_\omega(\Sigma)$, G is consistent, determines* nat *and* bool *and proves* $\mathsf{NStdAx}_0(\Sigma)$. *Then $I =_{df} \boldsymbol{T}(\Sigma, G, \mathsf{CondSUEq}_\omega(\Sigma))$ is an initial N-standard model of G. Furthermore, if t_1, t_2 are two closed Σ-terms of the same sort, then the following are equivalent:*

(i) t_1 and t_2 have the same value in I,

(ii) t_1 and t_2 have the same value in all N-standard models of G,

(iii) $t_1 = t_2$ is provable from G in $\mathsf{CondSUEq}_\omega(\Sigma)$,

(iv) $t_1 = t_2$ is provable from G in $\mathsf{FOL}_\omega(\Sigma)$.

Proof: By the N-standardness Lemma, I is N-standard. Again, the main thing is to show that I satisfies closed substitution instances of axioms of G. By definition, for any *closed equation P*:

$$I \models P \iff G \vdash P \tag{$*$}$$

where '$\vdash$' here means provability in $\mathsf{CondSUEq}_\omega$. Further, by use of the $\forall_\omega R$ rule, the same holds for any *closed SU equation R*:

$$I \models R \iff G \vdash R.$$

For suppose $R \equiv \forall \mathsf{z} P(\mathsf{z})$, where $P(\mathsf{z})$ is an equation. Then

$$
\begin{aligned}
I \models \forall \mathsf{z} P(\mathsf{z}) &\iff \text{for all } n, \; I \models P(\bar{n}) \\
&\iff \text{for all } n, \; G \vdash P(\bar{n}) \qquad \text{by } (*) \\
&\iff G \; \vdash \forall \mathsf{z} P(\mathsf{z}) \qquad\qquad \text{by } \forall_\omega R
\end{aligned}
$$

The rest of the proof follows the pattern of Theorems 1, 2 and 3.   □

Remarks. (1) As before, the assumption in the theorem that G determines nat and bool can be replaced by the assumption that Σ is strictly N-standard.

(2) (*Completeness and conservativity.*) Once again, the implication $(ii) \Rightarrow (iii)$ can be viewed as a completeness theorem, and $(iv) \Rightarrow (iii)$ as a conservativity theorem.

7.6   Open term algebras

So far (Theorems 1, 2, 3 and 4) we have concentrated on closed term algebras. We could also formulate our results in a more general setting, namely, with term algebras constructed from open terms, *i.e.*, terms containing free variables (from a given set X).

The problem here is that with open terms (an analogy of) the N-Standardness Lemma (§7.2) will fail in general. However, under a certain syntactic condition (the "N-term condition" below), a version of this Lemma can still be formulated.

First we need some definitions and notation. Given a signature Σ, and a set $X \subseteq \boldsymbol{Var}(\Sigma)$, let $\boldsymbol{T}(\Sigma, X)$ be the set of Σ-terms in X, *i.e.*, Σ-terms containing variables from X only. In particular, for $X = \emptyset$, we have the set of closed Σ-terms $\boldsymbol{T}(\Sigma) = \boldsymbol{T}(\Sigma, \emptyset)$.

Given a first-order Σ-theory T and formalism $\mathcal{F}$ which is valid for $\boldsymbol{Alg}(\Sigma, T)$, let $\boldsymbol{T}(\Sigma, X, T, \mathcal{F})$ be the Σ-*term algebra* formed from $\boldsymbol{T}(\Sigma, X)$ by identifying terms provably equal from T in $\mathcal{F}$. (The closed term algebra $\boldsymbol{T}(\Sigma, T, \mathcal{F})$ considered above corresponds to the special case $X = \emptyset$).

The algebra $I =_{df} \boldsymbol{T}(\Sigma, X, T, \mathcal{F})$ is *free for T over X*. This means that for every model A of T, and every assignment $\rho : X \to A$ of elements of A to variables in X (of the same sort), there is a *unique Σ-homomorphism* $h : I \to A$ such that $h \upharpoonright X\rho$. (This reduces to initiality in $\boldsymbol{Alg}(\Sigma, T)$ when $X = \emptyset$.)

Note that I need not itself be a model of T. However, this will be the case, provided T satisfies certain syntactic conditions (*e.g.*, if T is a conditional equational theory; *cf.* Theorem 1 above).

Again, assuming that Σ is N-standard, we are interested in the question whether I is N-standard. A useful criterion in this connection is the following syntactic condition on Σ and X:

N-term Condition for (Σ, X). *No Σ-term of sort* nat *or* bool *contains any variables from X.*

Remarks. (1) The N-term condition for (Σ, X) is trivially satisfied when $X = \emptyset$.

(2) When Σ is strictly N-standard, it is equivalent to the condition:

> *there are no variables in X of sort* nat *or* bool.

This follows from Remark 3 in §1.5.

Now the theory given above, and specifically Theorems 1 to 4, can be generalised to the case of open term models $\boldsymbol{T}(\Sigma, X, T, \mathcal{F})$, where (Σ, X) satisfies the N-term condition. First, the N-standardness lemma becomes:

N-Standardness Lemma^{X}. *Suppose that (Σ, X) satisfies the N-term condition. Suppose further that in $\mathcal{F}$*

(i) T is consistent,

(ii) T determines nat *and* bool, *and*

(iii) T proves $\mathsf{NStdAx}_0(\Sigma)$.

Then $\boldsymbol{T}(\Sigma, X, T, \mathcal{F})$ is N-standard.

Next, the strict N-standardness lemma becomes (using Remark 2 above):

Strict N-Standardness Lemma^X. *Suppose Σ is strictly N-standard, and there are no variables in X of sort* nat *or* bool. *Suppose also $\mathcal{F}$ is at least as strong as* CondEq(Σ), *and in $\mathcal{F}$*

(i) *T is consistent, and*

(ii) *T proves* NStdAx(Σ).

Then $\boldsymbol{T}(\Sigma, X, T, \mathcal{F})$ is N-standard.

Consider next, for example, Theorem 2. This can be reformulated as follows.

Theorem 2^X. *Suppose (Σ, X) satisfies the N-term condition. Let E be a conditional equational theory over Σ. Suppose that in* CondEq(Σ), *E is consistent, determines* nat *and* bool, *and proves* NStdAx(Σ). *Then $I =_{df} \boldsymbol{T}(\Sigma, X, E, $ CondEq$(\Sigma))$ is an N-standard model of E, which is free for E over X. Furthermore, if t_1, t_2 are two terms in $\boldsymbol{T}(\Sigma, X)$ of the same sort, then the following are equivalent:*

(i) *t_1 and t_2 have the same value in I,*

(ii) *t_1 and t_2 have the same value in all N-standard models of E,*

(iii) *$t_1 = t_2$ is provable from E in* CondEq(Σ),

(iv) *$t_1 = t_2$ is provable from E in* FOL$(\Sigma) + $ Ind(Σ).

The strict N-standardness Lemma^X, and Theorem 2^X, will be used in Section 9.

7.7 Reducing conditional BU to conditional equational specifications

We re-consider the work of §3.3 from the viewpoint of initial algebra specifications.

Theorem 5 (BU elimination for initial algebra specifications). *Let F be a conditional BU equational theory over Σ. Then there is an expansion Σ' of Σ and a conditional equational theory E' over Σ' which is equivalent to F (relative to N-standard models) in the sense that:*

(i) *if A is an N-standard Σ-model of F, then it has a Σ'-expansion which is a N-standard model of E';*

(ii) *if $A \cong \boldsymbol{InitNStdAlg}(\Sigma, F)$ then it has a unique (up to Σ'/Σ-isomorphism) Σ'-expansion A' such that $A' \cong \boldsymbol{InitNStdAlg}(\Sigma', E')$;*

(iii) *if A' is an N-standard Σ'-model of E', then its Σ-reduct A is an N-standard model of F; and if $A' \cong \boldsymbol{InitNStdAlg}(\Sigma', E')$ then $A \cong \boldsymbol{InitNStdAlg}(\Sigma, F)$.*

If F contains q occurrences of BU quantifiers, then Σ' expands Σ by one new sort and q new function symbols. Moreover, if F is finite, with e axioms (say), then so is E', with $e + 4q$ axioms.

Proof: The idea, again, is to incorporate in the signature, for each BU quantifier occurring in F, a *characteristic function* for that quantifier. The problem with adjoining a boolean-valued function symbol $f : $ nat $\times u \rightarrow$ bool satisfying $(**)$ in the BU elimination theorem

in §3.3, is in the case that A is an initial N-standard model of F. In order that its Σ'-expansion A' be *N-standard*, the value of $\mathsf{f}(n, x)$ must be either true or false for every value of the arguments n, x. Furthermore, in order that A' also be *initial*, the Σ-homomorphism h from A to every N-standard model B of F must be extendible to a Σ'-homomorphism h' from A' to the Σ'-expansion B' of B. However, the rhs of $(**)$ in §3.3 will hold "more often" in B than in A (since B is a homomorphic image of A), with a corresponding *change* in the value of $\mathsf{f}(n, x)$ from false to true! Hence h cannot, in general, be extended as desired. (Making f a 0,1-valued function will cause exactly the same problem.)

We therefore adjoin a special sort D for the range of such functions f, with a constant d which takes the place of 'true' in $(**)$ in §3.3. (The point is that when the condition on the rhs of $(**)$ fails, $\mathsf{f}(n, x)$ is not "forced" to equal anything else at all.) Now for each BU quantifier as in $(*)$ of §3.3, *adjoin* to the signature the function symbol

$$\mathsf{f} : \mathsf{nat} \times u \;\rightarrow\; \mathsf{D},$$

and adjoin the axioms formed from $(***)$ and $(****)$ in §3.3 by replacing 'true' by 'd' throughout. In this way we *replace F* by a conditional equational theory E' in Σ', with the stated properties. $\quad\square$

Remark. If A is an N-standard model of F, then its N-standard Σ'-expansion A' modelling E', given by part (i) of the theorem, is not (in general) uniquely determined. However, the added condition of initiality (on A and A') determines A' uniquely.

8 Initiality-preserving operators on N-standard algebras

In this section we combine the theory of Section 5 ("computability $\implies$ algebraic specifiability") with the initial algebra theory of Section 7.

8.1 Initiality preserving operators and the HEP

Assume now (as in §3.1) that Σ' and Σ'' are N-standard signatures with $\Sigma \subset \Sigma' \subset \Sigma''$, and $\Phi : \textbf{\textit{NStdAlg}}\,(\Sigma) \rightarrow \textbf{\textit{NStdAlg}}(\Sigma')$ is an expanding operator over Σ. Recall Definitions 5 and 7 in §3.1.

Definition 1. Φ is *initiality preserving (w.r.t. Σ and Σ')* iff for all $\mathbb{K} \subseteq \textbf{\textit{NStdAlg}}\,(\Sigma)$ and $A \in \textbf{\textit{NStdAlg}}\,(\Sigma)$, A is initial in $\mathbb{K}$ iff A^Φ is initial in $\mathbb{K}^\Phi$.

Lemma 1. *Suppose Φ is initiality preserving, and (Σ', T') specifies Φ uniformly over Σ. Then for any Σ-theory T and N-standard Σ-algebra A,*

$$A \cong \textbf{\textit{InitNStdAlg}}(\Sigma, T) \;\Longleftrightarrow\; A^\Phi \cong \textbf{\textit{InitNStdAlg}}(\Sigma', T + T').$$

Lemma 2. *Suppose* $\Phi(A) = \Psi(A)\,|_{\Sigma'}$ *for all* $A \in \boldsymbol{NStdAlg}\,(\Sigma)$, *where*

$$\Psi : \boldsymbol{NStdAlg}\,(\Sigma) \to \boldsymbol{NStdAlg}(\Sigma'')$$

is an expanding operator which is initiality preserving w.r.t. Σ *and* Σ''. *Then* Φ *is initiality preserving, and for any* Σ''-*theory* T'' *and N-standard* Σ-*algebra* A, *if* (Σ'', T'') *specifies* Ψ *uniformly over* Σ, *then* (Σ'', T'') *specifies* Φ *uniformly over* Σ *with hidden sorts and/or functions; and for any* Σ-*theory* T *and N-standard* Σ-*algebra* A,

$$
\begin{aligned}
A \cong \boldsymbol{InitNStdAlg}(\Sigma, T) &\iff A^\Psi \cong \boldsymbol{InitNStdAlg}(\Sigma'', T + T'') \\
&\iff A^\Phi \cong \boldsymbol{InitNStdAlg}(\Sigma'', T + T'')\,|_{\Sigma'} \\
&\iff A^\Phi \cong \boldsymbol{Init}\big(\Sigma',\, \boldsymbol{NStdAlg}(\Sigma'', T + T'')\,|_{\Sigma'}\,\big).
\end{aligned}
$$

Proof: From Lemma 1. □

Definition 2. Φ *has the* homomorphism extension property (HEP) (w.r.t. Σ and Σ') *iff every homomorphism* $h : A \to B$ *between N-standard* Σ-*algebras can be* extended uniquely *to a homomorphism* $h^\Phi : A^\Phi \to B^\Phi$ *between their images under* Φ.

Lemma 3. *If* Φ *has the HEP, then* Φ *is initiality preserving.*

We will apply the above theory to three cases: array specifications in §8.2, and specifications for PR and μPR* computable functions in §8.3 and §8.4 respectively.

8.2   Initial algebra specification of array algebras

Recall the array specification $(\Sigma^*, \mathsf{ArrAx}(\Sigma))$ defined in §3.2.

Lemma 1. *The array construction* $A \mapsto A^*$ *(§1.6) has the HEP, and (hence) is initiality preserving.*

Lemma 2. *For any N-standard* Σ-*algebra* A *and* Σ-*theory* T:

$$A \cong \boldsymbol{InitNStdAlg}(\Sigma, T) \iff A^* \cong \boldsymbol{InitNStdAlg}(\Sigma^*, T + \mathsf{ArrAx}(\Sigma)).$$

Proof: By §8.1, Lemma 1, and §3.2, Theorem 1. □

Of particular interest is the case that T is a *conditional BU equational* theory:

Theorem 1. *If a* Σ-*algebra* A *has an initial N-standard algebra specification by a set of conditional BU equations, then so does* A^*. *Moreover, if the specification for* A *is finite, with* e *axioms (say), then so is that for* A^*, *with at most* $e + 8s$ *axioms, where* s *is the number of sorts in* Σ.

Next, from the BU elimination theorem for initial algebras (Theorem 5 in Section 7) we can reduce such a specification for A^* further to one with *conditional equations* only.

Theorem 2. *If a Σ-algebra A has an initial N-standard algebra specification by a set of conditional equations, then so does A^* (with hidden sorts and functions). Moreover, if the specification for A is finite, with e axioms (say), then so is that for A^*, with at most $e + 12s$ axioms, where s is the number of sorts in Σ.*

Proof: First apply Theorem 1 (or Lemma 2) above. Then replace the equality axiom for s^* in $\mathsf{ArrAx}(\Sigma)$, which is a conditional BU Σ^*-equation (§3.2), by a conditional Σ^*-equation, for each Σ-equality sort s other than $\mathtt{nat}$, by BU elimination (Theorem 5 in §7.7, applied to Σ^*). $\square$

8.3   Initial algebra specifications for PR computable functions

Now we apply the above theory to the results in §5.1.

Lemma 1. *For each $PR(\Sigma)$ derivation α, the operator $(**)$ (§5.1) has the HEP, and is (therefore) initiality preserving. Hence the operator $(*)$ is initiality preserving.*

Proof: By structural induction on α. $\square$

Hence, by Theorem 1 in Section 5 and Lemma 2 in §8.1:

Lemma 2. *For each $PR(\Sigma)$ derivation α, and for each N-standard Σ-algebra A and Σ-theory T:*

$$A \cong \textbf{\textit{InitNStdAlg}}(\Sigma, T) \iff (A, \mathsf{g}_\alpha^A, \mathsf{f}_\alpha^A) \cong \textbf{\textit{InitNStdAlg}}(\Sigma_\alpha, T + E_\alpha)$$
$$\iff (A, \mathsf{f}_\alpha^A) \cong \textbf{\textit{InitNStdAlg}}(\Sigma_\alpha, T + E_\alpha)\,|\,{}_{\Sigma_\mathsf{f}}$$
$$\iff (A, \mathsf{f}_\alpha^A) \cong \textbf{\textit{Init}}\big(\Sigma_\mathsf{f}, \textbf{\textit{NStdAlg}}(\Sigma_\alpha, T + E_\alpha)\,|\,{}_{\Sigma_\mathsf{f}}\big).$$

Here $\Sigma_\mathsf{f} = \Sigma \cup \{\mathsf{f}_\alpha\}$. (Remember, $\Sigma_\alpha = \Sigma \cup \{\mathsf{g}_\alpha, \mathsf{f}_\alpha\}$, where g_α is the list of auxiliary functions of α.) Of particular interest is the case that T is a *conditional equational* theory:

Theorem 3. *Let f be a PR function on a Σ-algebra A. If A has an initial N-standard algebra specification by a set of conditional equations, then so does (A, f) (with hidden functions).*

8.4   Initial algebra specifications for μPR^* computable functions

We turn to μPR^* computability (§5.2). The problem here (as noted in §5.2) is that even if the computed function is total, the auxiliary functions need not be. However, by applying the totality lemma (§5.2), we are able restrict our attention to total derivations.

Lemma 1. *For each $\mu PR^*(\Sigma)$ derivation α and each N-standard Σ-algebra A on which f_α^A is total, the operator $(***)$ (§5.2) has the HEP, and is (therefore) initiality preserving. Hence the operator $(*)$ (§5.1) is initiality preserving.*

Proof: By structural induction on α. $\square$

Hence, by Theorem 2 in Section 5 and Lemma 2 in §8.1, we have:

Lemma 2. *For each $\mu PR^*(\Sigma)$ derivation α, each N-standard Σ-algebra A on which f_α^A is total, and each Σ-theory T:*

$$A \cong \mathbf{\mathit{InitNStdAlg}}(\Sigma, T) \iff$$
$$(A, \mathsf{f}_\alpha^A) \cong \mathbf{\mathit{Init}}\big(\Sigma_{\mathsf{f}}, \mathbf{\mathit{NStdAlg}}(\Sigma_\alpha^*, T + \mathsf{ArrAx}(\Sigma) + F_{\hat{\alpha}}) \,|\, _{\Sigma_{\mathsf{f}}}\big).$$

where $\hat{\alpha}$ is the total derivation for f_α given by the totality lemma, and $F_{\hat{\alpha}}$ is the conditional BU specification for $\hat{\alpha}$.

Here, as before, $\Sigma_{\mathsf{f}} = \Sigma \cup \{\mathsf{f}_\alpha\}$. Of particular interest are the two cases that T is a *conditional BU equational* theory, and a *conditional equational* theory. First, assuming the former:

Theorem 4. *Let f be a total μPR^* function on a Σ-algebra A. If A has an initial N-standard algebra specification (Σ, F), where F is a set of conditional BU equations, then likewise (A, f) has such a specification $(\Sigma_{\mathsf{f}}, F^f)$ with hidden sorts and functions, where F^f is also a set of conditional BU equations. Moreover, F^f can be obtained by adjoining to F an instantiation $F^U(\bar{k})$ of some universal conditional BU equational specification $F^U(\mathsf{z})$, which depends only on Σ and the type of f.*

The universal specification $F^U(\mathsf{z})$ in this theorem is obtained as in Theorem 3 in Section 5.

Finally, by assuming T in Lemma 2 is a conditional equational theory, and applying Theorem 4 above and then BU elimination for initial algebras (Theorem 5 in Section 7):

Theorem 5. *Let f be a total μPR^* function on a Σ-algebra A. If A has an initial N-standard algebra specification (Σ, E), where E is a set of conditional equations, then likewise (A, f) has such a specification $(\Sigma_{\mathsf{f}}, E^f)$ with hidden sorts and functions, where E^f is also a set of conditional equations. Moreover, E^f can be obtained by adjoining to E an instantiation $E^U(\bar{k})$ of some universal conditional equational specification $E^U(\mathsf{z})$, which depends only on Σ and the type of f.*

9 Computability of algebraically specifiable functions

In this section we prove (partial) converses to the results of Section 5. First we need a definition.

Definition (Strong specifiability). Let $\mathbb{K}$ be a Σ-class, let $\Sigma' \supseteq \Sigma \cup \{\mathsf{f}\}$ and let T be a Σ'-theory. We say that T *strongly specifies* a family $\{\, f^A \mid A \in \mathbb{K} \,\}$ (possibly with hidden sorts and/or functions) iff
(i) T specifies $\{\, f^A \mid A \in \mathbb{K} \,\}$, and further
(ii) for every $A, B \in \mathbb{K}$ with $B \preceq A$, $f^B = f^A \restriction B$.

(Here $f^A \restriction B$ denotes the restriction of f^A to B.)

The significance of this concept is seen by rephrasing it in either of the following two ways.

48

Lemma 1. *Let $\mathbb{K}$ be a Σ-class, let $\Sigma' \supseteq \Sigma \cup \{\mathsf{f}\}$ and let T be a Σ'-theory. T strongly specifies a family $\{\, f^A \mid A \in \mathbb{K} \,\}$ (possibly with hidden sorts and/or functions) iff*
(i) T specifies $\{\, f^A \mid A \in \mathbb{K} \,\}$, and further
(ii') for every $A, B \in \mathbb{K}$ with $B \preceq A$, B is closed under f^A.

Lemma 2. *Let $\mathbb{K}$ be a Σ-class which is closed under finitely generated subalgebras, let $\Sigma' \supseteq \Sigma \cup \{\mathsf{f}\}$ and let T be a Σ'-theory. T strongly specifies a family $\{\, f^A \mid A \in \mathbb{K} \,\}$ (possibly with hidden sorts and/or functions) iff*
(i) T specifies $\{\, f^A \mid A \in \mathbb{K} \,\}$, and further
(ii'') for every $A \in \mathbb{K}$ and every finitely generated $B \preceq A$, B is closed under f^A.

We consider algebras and functions specified by conditional equational theories. We have to assume now that these theories have *effective axiomatisations*: that the axioms are finite, for example, or at least recursively enumerable.

We will also make use of Theorem 2^X in §7.6. Recall the remarks preceding the theorem there, that the N-term condition for (Σ, X) follows from either (i) $X = \emptyset$; or (ii) strict N-standardness of Σ, together with X containing no variables of sort nat or bool.

We will prove two theorems, making each of these assumptions in turn.

9.1 Computability of specifiable function on minimal algebras

We first consider a partial converse, using Remark 1 on the N-term condition (§7.6), that is, restricting our attention to minimal models (*i.e.*, models in which every element is named by a closed term). We use the notation $\boldsymbol{MinNStdAlg}(\Sigma, T)$ for the set of minimal N-standard Σ-models of a theory T.

Theorem 1. *Suppose Σ is N-standard. Let E be an r.e. conditional equational Σ-theory which in $\mathsf{CondEq}(\Sigma)$ is consistent, determines nat and bool and proves $\mathsf{NStdAx_0}(\Sigma)$. Suppose $\Sigma' \supseteq \Sigma \cup \{\mathsf{f}\}$, and let E' be an r.e. conditional equational Σ'-theory which strongly specifies $\{\, f^A \mid A \in \boldsymbol{MinNStdAlg}(\Sigma, E) \,\}$ (possibly with hidden sorts and/or functions). Assume also that $E + E'$ determines nat and bool, and is conservative over E, in $\mathsf{CondEq}(\Sigma)$, and also that all sorts of $\boldsymbol{dom}(f)$ other than bool are equality sorts. Then f^A is uniformly μPR^* computable over $A \in \boldsymbol{MinNStdAlg}(\Sigma, E)$.*

Proof: We will describe a pseudo-$\boldsymbol{While^*}(\Sigma)$ algorithm for computing f^A uniformly over minimal N-standard Σ-models A of E. Suppose $\mathsf{f} : u \to s$, where $u = s_1 \times \cdots \times s_n$. In general, some of the s_i are nat or bool, and the others not. Suppose (w.l.o.g.) that for some $m < n$, sorts $s_{m+1}, \ldots, s_n$ are all either nat or bool, and sorts $s_1, \ldots, s_m$ are not. Write $u = v \times w$ where $v = s_1 \times \cdots \times s_m$ and $w = s_{m+1} \times \cdots \times s_n$. By assumption, sorts $s_1, \ldots, s_m$ are equality sorts.

For any $A \in \boldsymbol{MinNStdAlg}(\Sigma, E)$, we will show how to compute

$$f^A : A^u \to A_s.$$

Choose a tuple $k = (k_1, \ldots, k_{n-m}) \in A^w$ (of naturals and truth values), and consider the function

$$f^A_k =_{df} f(\,\cdot\,, k) : A^v \to A_s.$$

ACM Transactions on Computational Logic, Vol. TBD, No. TBD, TBD TBD.

We will show how to compute f_k^A uniformly in the (numerical and boolean) parameters k.

Let $I = \boldsymbol{T}(\Sigma, E, \mathsf{CondEq}(\Sigma))$ and $J = \boldsymbol{T}(\Sigma', E + E', \mathsf{CondEq}(\Sigma'))$ (recall the definitions in §7.1). By the N-Standardness Lemma (and the conservativity assumption for $E + E'$ over E), both I and J are N-standard. (Below we denote elements of these algebras by '$[t]$', *i.e.*, suitable equivalence classes of terms t, or tuples thereof. We also write $\bar{k}$ for the tuple of numerals and/or truth constants corresponding to k.)

Note that the identity mapping on $\boldsymbol{T}(\Sigma)$ induces a Σ-homomorphism

$$\iota_I : \ I \ \to \ J \mid_\Sigma.$$

By conservativity of $E + E'$ over E, ι_I is injective. Hence $I \preceq J \mid_\Sigma$.

Further, the function f^J specified by E' on $J \mid_\Sigma$ is clearly the same as that defined "naturally" on J by $f^J([t]) = [\mathsf{f}(t)]$. By the strong specification assumption,

$$f^I = f^J \restriction I.$$

Hence for any closed Σ-term t_0,

$$f_k^J([t_0]) \ = \ f_k^I([t_0]) \ = \ [t]$$

for some closed Σ-term t. By definition of J, this means that the equation

$$\mathsf{f}(t_0, \bar{k}) \ = \ t \tag{$*$}$$

is *provable* from $E + E'$ in $\mathsf{CondEq}(\Sigma')$.

Now take any $A \in \boldsymbol{MinNStdAlg}(\Sigma, E)$, and any $a \in A^v$. Since A is minimal, there is a tuple of closed Σ-terms $t_0 : v$ such that $t_0^A = a$. By Theorem 2 of Section 6 applied to Σ', there is a Σ'-homomorphism

$$h : \ J \ \to \ (A, f^A, \ldots)$$

with $h([t_0]) = a$. Hence, since $(*)$ holds in J, it also holds in (the Σ'-expansion of) A, with 'f' interpreted as f^A.

This suggests the following algorithm for f_k^A with A minimal. With *inputs* $a \in A^v$: first generate all (Gödel numbers of) tuples of closed Σ-terms of type v, until you find a tuple t_0 with $t_0^A = a$. (This is where we use computability of equality on type v.) Then *generate* all Gödel numbers of theorems of $E + E'$ until you find one of the form $\ulcorner \mathsf{f}(t_0) = t \urcorner$, for some closed Σ-term t. Then the *output* is t^A.

The search is effective in the *term evaluation* function for closed Σ-terms in A, by recursive enumerability of E and E'. Further, since term evaluation is PR* computable ([TZ00, §4]), this algorithm can be formalised as a $\mu\mathrm{PR}^*(\Sigma)$ derivation for f^A, as desired. $\quad\square$

Remark. The assumption that the sorts of $\boldsymbol{dom}(f)$ are equality sorts can clearly be weakened to the assumption that equality is (uniformly over $\boldsymbol{MinNStdAlg}(\Sigma, E)$) computable on these sorts.

9.2 Computability of specifiable function in strictly N-standard algebras

We consider a second partial converse, using Remark 2 on the N-term condition, *i.e.*, no free variables of sort nat or bool, plus strict N-standardness.

Theorem 2. *Suppose Σ is strictly N-standard. Let E be an r.e. conditional equational Σ-theory which in $\mathsf{CondEq}(\Sigma)$ is consistent and proves $\mathsf{NStdAx}(\Sigma)$. Suppose $\Sigma' \supseteq \Sigma \cup \{\mathsf{f}\}$ is also strictly N-standard and proves $\mathsf{NStdAx}(\Sigma')$. Let E' be an r.e. conditional equational Σ'-theory which strongly specifies $\{\, f^A \mid A \in \boldsymbol{NStdAlg}(\Sigma, E)\,\}$ (possibly with hidden sorts and/or functions). Assume also that $E + E'$ is conservative over E in $\mathsf{CondEq}(\Sigma')$. Then f^A is uniformly μPR^* computable over $A \in \boldsymbol{NStdAlg}(\Sigma, E)$.*

Proof: We will describe a pseudo-***While****$^*(\Sigma)$ algorithm for computing f^A uniformly over $A \in \boldsymbol{NStdAlg}(\Sigma, E)$. Suppose $\mathsf{f} : u \to s$, where $u = s_1 \times \cdots \times s_n$. In general, some of the s_i are nat or bool, and the others not. Suppose (w.l.o.g.) that for some $m < n$, sorts $s_{m+1}, \ldots, s_n$ are all either nat or bool, and sorts $s_1, \ldots, s_m$ are not. Write $u = v \times w$ where $v = s_1 \times \cdots \times s_m$ and $w = s_{m+1} \times \cdots \times s_n$.

For any $A \in \boldsymbol{NStdAlg}(\Sigma, E)$, we will show how to compute

$$f^A : \ A^u \ \to \ A_s.$$

Choose a tuple $k = (k_1, \ldots, k_{n-m}) \in A^w$ (of naturals and truth values), and consider the function

$$f_k^A =_{df} f(\,\cdot\,, k) : \ A^v \ \to \ A_s.$$

We will show how to compute f_k^A uniformly in the (numerical and boolean) parameters k.

Choose a tuple of variables $\mathbf{x} : v$ (*i.e.*, of the same product type as a). Let $I = \boldsymbol{T}(\Sigma, \mathbf{x}, E, \mathsf{CondEq}(\Sigma))$ and $J = \boldsymbol{T}(\Sigma', \mathbf{x}, E + E', \mathsf{CondEq}(\Sigma'))$ (recall the definitions in §7.6). By the strict N-standardness Lemma^X (§7.6), both I and J are N-standard.

Note that the identity mapping on $\boldsymbol{T}(\Sigma, \mathbf{x})$ induces a Σ-homomorphism

$$\iota_I : \ I \ \to \ J \,|\, _\Sigma.$$

By conservativity of $E + E'$ over E, ι_I is injective. Hence $I \preceq J \,|\,_\Sigma$.

Further, the function f^J specified by E' on $J \,|\,_\Sigma$ is clearly the same as that defined naturally on J by $f^J([t]) = [\mathsf{f}(t)]$. By the strong specification assumption,

$$f^I \ = \ f^J \restriction I.$$

Hence

$$f_k^J(\mathbf{x}) \ = \ f_k^I(\mathbf{x}) \ = \ [t]$$

for some $t \in \boldsymbol{T}(\Sigma, \mathbf{x})$. By definition of J, this means that the equation

$$\mathsf{f}(\mathbf{x}, \bar{k}) \ = \ t \tag{$*$}$$

is *provable* from $E + E'$ in $\mathsf{CondEq}(\Sigma')$.

Now take any $A \in \textbf{NStdAlg}(\Sigma, E)$, and any $a \in A^v$. By Theorem 2^X applied to Σ', there is a Σ'-homomorphism

$$ h : \ J \ \rightarrow \ (A, \, f^A, \, \ldots) $$

where $h(\mathtt{x}) = a$. Hence, since $(*)$ holds in J, it also holds in (the Σ'-expansion of) A, with 'f' interpreted as f^A and a assigned to $\mathtt{x}$.

This suggests the following algorithm for f_k^A. With *inputs* $a \in A^v$: *generate* all Gödel numbers of theorems of $E + E'$ until you find one of the form $\ulcorner \mathtt{f}(\mathtt{x}) = t \urcorner$, for some Σ-term t (in $\mathtt{x}$). This search is effective, by recursive enumerability of E and E'. Then the *output* is the *evaluation of the term t* in A with a assigned to $\mathtt{x}$.

Since term evaluation is PR* computable [TZ00, §4], this algorithm can be formalised as a μPR$^*(\Sigma)$ derivation for f^A, as desired. □

Remarks.

(1) The above algorithm gives, for each tuple of numerical and boolean arguments k, a *fixed* term $t \in \textbf{T}(\Sigma, \mathtt{x})$ as the value of $f_k^A(a)$ for all $A \in \textbf{NStdAlg}(\Sigma, E)$ and all $a \in A^v$.

(2) Theorems similar to Theorems 1 and 2 can be formulated for conditional BU equational theories and specifications, using a variation of Theorem 3 (instead of Theorem 2) in Section 7.

9.3 Significance of strong specifiability; Equivalence of specifiability and computability

We want to combine some of the above results into an equivalence result between computability and specifiability.

Note that by the Locality Theorem for **While** computations [TZ00, §2.8], if f is μPR* computable on an algebra A, then any subalgebra of A is closed under f. This suggests the following formulations for equivalence theorems, which are simple consequences of the above theorems and the lemmas on strong specifiability at the beginning of this section.

We give one formulation (Theorem 3) for minimal algebras (*cf.* Theorem 1), and another (Theorem 4) for strictly N-standard algebras (*cf.* Theorem 2).

Theorem 3. *Suppose Σ is N-standard. Let E be an r.e. conditional equational Σ-theory, which in $\mathsf{CondEq}(\Sigma)$ is consistent, determines nat and bool and proves $\mathsf{NStdAx}_0(\Sigma)$. Let $\boldsymbol{f} = \langle f^A \mid A \in \textbf{MinNStdAlg}(\Sigma, E) \rangle$ be a family of functions on $\textbf{MinNStdAlg}(\Sigma, E)$. Assume that all sorts of $\boldsymbol{dom}(\boldsymbol{f})$ other than bool are equality sorts. Then the following are equivalent:*

(i) $\boldsymbol{f}$ is μPR* computable uniformly on $\textbf{MinNStdAlg}(\Sigma, E)$;

(ii) $\boldsymbol{f}$ is strongly specifiable uniformly on $\textbf{MinNStdAlg}(\Sigma, E)$, with hidden sorts and functions, by a finite set of conditional equations which (together with E) is conservative over E in $\mathsf{CondEq}(\Sigma)$.

Theorem 4. *Suppose Σ is strictly N-standard. Let E be an r.e. conditional equational Σ-theory, which in* $\mathsf{CondEq}(\Sigma)$ *is consistent and proves* $\mathsf{NStdAx_0}(\Sigma)$. *Let* $\boldsymbol{f} = \langle f^A \mid A \in \boldsymbol{NStdAlg}(\Sigma, E) \rangle$ *be a family of functions on* $\boldsymbol{NStdAlg}(\Sigma, E)$. *Then the following are equivalent:*

(i) $\boldsymbol{f}$ *is μPR^* computable uniformly on* $\boldsymbol{NStdAlg}(\Sigma, E)$;

(ii) $\boldsymbol{f}$ *is strongly specifiable uniformly on* $\boldsymbol{NStdAlg}(\Sigma, E)$, *with hidden sorts and functions, by a finite set of conditional equations which (together with E) is conservative over E in* $\mathsf{CondEq}(\Sigma)$, *and such that the signature of these equations is also strictly N-standard.*

Remark (Herbrand-Gödel computability on $\mathbb{N}$). The above theorem generalises the classical equivalence result on $\mathcal{N}$ [Kle52].

10 Concluding remarks and future directions

10.1 Computation on Topological Data Types

We have extended the theory of algebraic specifications from the world of countable computable algebras to that of all algebras, and especially metric algebras, by means of abstract computability theory. Topological data types and algebraic specifications play a fundamental role in many areas of computing, including semantics and scientific computation.

Our main theorems concern the transformation of abstract algorithms to algebraic specifications and provide some basic techniques for the theory of specifying and verifying abstract computations. An obvious question is:

> *Under what circumstances can the conditional equations be replaced by equations in our theory?*

However, the converse results on the derivation of algorithms from algebraic specifications need strengthening to provide completeness or equivalence theorems. Improving results in the reverse direction is an important problem, as stated in the Introduction. There is much more to this topic than the results in Section 9. A key technical problem in this area is:

> *To develop general techniques for solving equations, conditional equations and other algebraic formulae in topological algebras.*

In semantics, for example, special cases of the problem are common. Semantic modelling makes heavy use of fixed-point equations. One thinks of the introduction of metric methods into semantics by M. Nivat (see [Niv79, AN80a, AN80b]), or their use in concurrency theory by De Bakker and others [dBZ82, dBR92, dBdV99]. Studies of the methods of equation solving in ultrametric algebras, including equivalence between metric, algebraic and domain-theoretic techniques, are in Stoltenberg-Hansen and Tucker [SHT88, SHT91, SHT93]; see also [SHLG94].

In scientific computation, numerical methods are concerned with obtaining computable solutions from differential and integral equations. Mathematical models of systems in the world are specified by sets of equations, from which algorithms are sought to compute their

solutions and hence to simulate the system. Our main theorems and examples in Section 5 show the opposite: if a system can be approximately simulated on a computer then there exist algebraic specifications that capture the system's behaviour. Such results seem to be new and, in our view, draw loci that help delimit the computability theory of physical systems. We conjecture that it is possible

To show that certain parts of the theory of numerical approximation of differential and integral equations are special instances of a general theory of algebraic specifications.

This is an exciting and difficult problem with many obvious applications.

Given the wealth of algorithms and theory in numerical methods, it seems to us that relatively little is known about the computational and logical scope and limits of equations, the classical mathematical methods of science. Progress in the area has awaited the creation of stable computation theories for topological data types. Over the past decade, computability theory for topological spaces and algebras has developed dramatically. Several general approaches have produced deep results and have been shown to be equivalent. Some approaches are

- metric spaces [Mos64],
- axiomatic computation structures [PER89],
- type two effectivity [Wei00],
- algebraic domain representability [SHT88, SHT95, Bla97],
- continuous domain representability [Eda97],
- abstract computability [TZ99, TZ00, BSS89]

The equivalence of the first five approaches is proved in [SHT99]. The equivalence of all these with the last one is proved in [Bra97, Bra99, TZ99, TZ01a].

However, this computability theory needs to be complemented by a logical theory which includes equation solving in topological algebras.

10.2   Theory of computable data types

The theory of algebraic specifications of computable (semicomputable, and cosemicomputable) data types contains many techniques for proving special properties of algebraic specifications, and showing the equivalence or non-equivalence of specification methods. Can some of these results be generalised? We believe the answer is yes, but not without much further work. Many results depend on special techniques of classical computability theory on the natural numbers. The theory for computable algebras uses representations by recursive algebras of numbers. It is possible to make a representation theory for topological algebras based on Baire space $\mathbb{N}^{\mathbb{N}}$ using the *type two effectivity* methods of Klaus Weihrauch [Wei00]. The use of the Diophantine Theorem for r.e. sets is more difficult: the theory of r.e. sets in abstract computability differs from the classical case, and no Diophantine Theorem is known (even for minimal algebras).

Since abstract computability theory is uniform over classes of algebras, our results on specifications are uniform, yielding parameterised specifications. As we have seen, this pro-

cess is surprisingly delicate because it leads to questions about standardness. In abstract computations it is natural to augment an algebra by basic data types such as booleans, naturals and finite arrays. These have an effect on the axiomatisations. There are other important additional types, of both theoretical and practical interest, that may be used to augment a given data type and are in need of a standard algebraic specification theory, including:

(i) infinite streams (necessary for developing the theory of interactive systems);

(ii) real numbers (necessary for developing the theory of metric algebras and normed linear spaces).

An attempt to extend the specification methods of this paper to both these data types, using infinitary equational specifications, is made in [TZ01b]

Finally, we note there are several other basic properties of specifications in need of investigation, especially term rewriting properties.

References

[AN80a]   A. Arnold and M. Nivat. Metric interpretations of infinite trees and semantics of nondeterministic recursive programs. *Theoretical Computer Science*, 11:181–205, 1980.

[AN80b]   A. Arnold and M. Nivat. The metric space of infinite trees, algebraic and topological properties. *Fundamenta Informaticae*, 4:445–476, 1980.

[Arn73]   V.I. Arnold. *Ordinary Differential Equations*. MIT Press, 1973.

[BCSS96] L. Blum, F. Cucker, M. Shub, and S. Smale. Complexity and real computation: A manifesto. *International Journal of Bifurcation and Chaos*, 6 (1):3–26, 1996.

[BCSS97] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer-Verlag, 1997.

[Bla97]   J. Blanck. Domain representability of metric spaces. *Annals of Pure & Applied Logic*, 83:225–247, 1997.

[Bra97]   V. Brattka. Order-free recursion on the real numbers. *Mathematical Logic Quarterly*, 43:216–234, 1997.

[Bra99]   V. Brattka. *Recursive and computable operations over topological structures*. Ph.d. thesis, FernUniversität Hagen, Fachbereich Informatik, Hagen, Germany, 1999. Informatik Berichte 255, FernUniversität Hagen, July 1999.

[BSS89]   L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: np-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21:1–46, 1989.

[BT80a]   J.A. Bergstra and J.V. Tucker. A characterisation of computable data types by means of a finite equational specification method. In J.W. de Bakker and

J. van Leeuwen, editors, *7th International Colloquium on Automata, Languages and Programming, Noordwijkerhout, The Netherlands, July 1980*, volume 85 of *Lecture Notes in Computer Science*, pages 76–90. Springer-Verlag, 1980.

[BT80b]   J.A. Bergstra and J.V. Tucker. A natural data type with a finite equational final semantics specification but no effective equational initial semantics specification. *Bulletin of the European Association for Theoretical Computer Science*, 11:23–33, 1980.

[BT82]   J.A. Bergstra and J.V. Tucker. The completeness of the algebraic specification methods for data types. *Information & Control*, 54:186–200, 1982.

[BT83]   J.A. Bergstra and J.V. Tucker. Initial and final algebra semantics for data type specifications: two characterization theorems. *SIAM Journal of Computing*, 12:366–387, 1983.

[BT87]   J.A. Bergstra and J.V. Tucker. Algebraic specifications of computable and semicomputable data types. *Theoretical Computer Science*, 50:137–181, 1987.

[BT95]   J.A. Bergstra and J.V. Tucker. Equational specifications, complete term rewriting and computable and semicomputable algebras. *Journal of the Association for Computing Machinery*, 42:1194–1230, 1995.

[dBdV99]   J.W. de Bakker and E. de Vink. *Control Flow Semantics*. The MIT Press, 1999.

[dBR92]   J.W. de Bakker and J.J.M.M. Rutten. *Ten Years of Concurrency Semantics*. World Scientific, 1992.

[dBZ82]   J.W. de Bakker and J.I. Zucker. Processes and the denotational semantics of concurrency. *Information and Control*, 54:70–120, 1982. Reprinted, with errata, in *Studies in Concurrency Semantics: Selected Papers of the Amsterdam Concurrency Group*, ed. J.W. de Bakker and J.J.M.M. Rutten, World Scientific Publishing Co. (1992), 28–80.

[Eda97]   A. Edalat. Domains for computation in mathematics, physics and exact real arithmetic. *Bulletin of Symbolic Logic*, 3:401–452, 1997.

[EM85]   H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification 1*, volume 6 of *EATCS Monographs*. Springer-Verlag, 1985.

[Eng93]   E. Engeler. *Algebraic Properties of Structures*. World Scientific Publishing Co., 1993.

[Gen69]   G. Gentzen. Investigations into logical deduction. In M.E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North Holland, 1969.

[GH78]   J.V. Guttag and J.J. Horning. The algebraic specification of abstract data types. *Acta Informatica*, 10:27–52, 1978.

[GH00]   T. Gärtner and G. Hotz. Recursive analytic functions of a complex variable. In *Computability and Complexity in Analysis: 4th Workshop, Swansea, September 2000*, pages 81–97. Technical Report 272-9/2000, FernUniversität Hagen, 2000.

[Grz55]   A. Grzegorczyk. Computable functions. *Fundamenta Mathematicae*, 42:168–202, 1955.

[Grz57]   A. Grzegorczyk. On the defintions of computable real continuous functions. *Fundamenta Mathematicae*, 44:61–71, 1957.

[Kle52]   S.C. Kleene. *Introduction to Metamathematics*. North Holland, 1952.

[Lac55]   D. Lacombe. *Extension de la notion de fonction récursive aux fonctions d'une ou plusieurs variables réelles*, I, II, III. *C.R. Acad. Sci. Paris*, 1955. 240:2470–2480, 241:13–14,151–153.

[Mal73]   A.I. Mal'cev. *Algebraic Systems*, volume 192 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, 1973.

[MG85]   J. Meseguer and J.A. Goguen. Initiality, induction and computability. In M. Nivat and J. Reynolds, editors, *Algebraic Methods in Semantics*, pages 459–541. Cambridge University Press, 1985.

[MM84]   B. Mahr and J.A. Makowsky. Characterizing specification languages which admit initial semantics. *Theoretical Computer Science*, 31:49–59, 1984.

[Mos64]   Y.N. Moschovakis. Recursive metric spaces. *Fundamenta Mathematicae*, 55:215–238, 1964.

[MT92]   K. Meinke and J.V. Tucker. Universal algebra. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 1, pages 189–411. Oxford University Press, 1992.

[Niv79]   M. Nivat. Infinite words, infinite trees, infinite computations. In J.W. de Bakker and J. van Leeuwen, editors, *Foundations of Computer Science III, part 2: Languages, Logic, Semantics*, volume 109 of *Mathematical Centre Tracts*, pages 3–52. Mathematical Centre, Amsterdam, 1979.

[PER89]   M.B. Pour-El and J.I. Richards. *Computability in Analysis and Physics*. Springer-Verlag, 1989.

[SHLG94]   Stoltenberg-Hansen, I. Lindström, and E. Griffor. *Mathematical Theory of Domains*. Cambridge University Press, 1994.

[SHT88]   V. Stoltenberg-Hansen and J.V. Tucker. Complete local rings as domains. *Journal of Symbolic Logic*, 53:603–624, 1988.

[SHT91]   V. Stoltenberg-Hansen and J.V. Tucker. Algebraic and fixed point equations over inverse limits of algebras. *Theoretical Computer Science*, 87:1–24, 1991.

[SHT93]   V. Stoltenberg-Hansen and J.V. Tucker. Infinite systems of equations over inverse limits and infinite synchronous concurrent algorithms. In J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Semantics: Foundations and Applications*, volume 666 of *Lecture Notes in Computer Science*, pages 531–562. Springer-Verlag, 1993.

[SHT95]   V. Stoltenberg-Hansen and J.V. Tucker. Effective algebras. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 4, pages 357–526. Oxford University Press, 1995.

[SHT99]   V. Stoltenberg-Hansen and J.V. Tucker. Concrete models of computation for topological algebras. *Theoretical Computer Science*, 219:347–378, 1999.

[Tak87]   G. Takeuti. *Proof Theory (2nd ed.)*. North Holland, 1987.

[Tuc80]   J.V. Tucker. Computing in algebraic systems. In F.R. Drake and S.S. Wainer, editors, *Recursion Theory, its Generalisations and Applications*, volume 45 of *London Mathematical Society Lecture Note Series*, pages 215–235. Cambridge University Press, 1980.

[TZ88]   J.V. Tucker and J.I. Zucker. *Program Correctness over Abstract Data Types, with Error-State Semantics*, volume 6 of *CWI Monographs*. North Holland, 1988.

[TZ91]   J.V. Tucker and J.I. Zucker. Projections of semicomputable relations on abstract data types. *International Journal of Foundations of Computer Science*, 2:267–296, 1991.

[TZ92]   J.V. Tucker and J.I. Zucker. Deterministic and nondeterministic computation, and horn programs, on abstract data types. *Journal of Logic Programming*, 13:23–55, 1992.

[TZ93]   J.V. Tucker and J.I. Zucker. Provable computable selection functions on abstract structures. In P. Aczel, H. Simmons, and S.S. Wainer, editors, *Proof Theory*, pages 277–306. Cambridge University Press, 1993.

[TZ99]   J.V. Tucker and J.I. Zucker. Computation by 'while' programs on topological partial algebras. *Theoretical Computer Science*, 219:379–420, 1999.

[TZ00]   J.V. Tucker and J.I. Zucker. Computable functions and semicomputable sets on many-sorted algebras. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 5, pages 317–523. Oxford University Press, 2000.

[TZ01a]   J.V. Tucker and J.I. Zucker. Abstract versus concrete models of computation on partial metric algebras. In preparation, 2001.

[TZ01b]   J.V. Tucker and J.I. Zucker. Infinitary initial algebra specifications for stream algebras. In W. Sieg, R. Sommer, and C. Talcott, editors, *Reflections: A Collection of Essays in Honor of Solomon Feferman*. Association for Symbolic Logic, 2001. To appear.

[Wec92]   W. Wechler. *Universal Algebra for Computer Scientists*, volume 25 of *EATCS Monographs*. Springer-Verlag, 1992.

[Wei00]   K. Weihrauch. *Computable Analysis: An Introduction*. Springer-Verlag, 2000.