Primitive Recursive Selection Functions for Existential Assertions over Abstract Algebras

Thomas Strahm
Institut für Informatik und angewandte Mathematik, Universität Bern, CH-3012 Bern, Switzerland
strahm@iam.unibe.ch

Jeffery Zucker
Department of Computing and Software, McMaster University, Hamilton, Ontario L8S 4K1, Canada
zucker@mcmaster.ca

(*October 8, 2007*)

Abstract. We generalize to abstract many-sorted algebras the classical proof-theoretic result due to Parsons, Mints and Takeuti that an assertion $\forall \mathtt{x}\, \exists \mathtt{y}\, P(\mathtt{x}, \mathtt{y})$ (where P is $\mathbf{\Sigma}^0_1$), provable in Peano arithmetic with $\mathbf{\Sigma}^0_1$ induction, has a primitive recursive selection function. This involves a corresponding generalization to such algebras of the notion of *primitive recursiveness*. The main difficulty encountered in carrying out this generalization turns out to be the fact that equality over these algebras may not be computable, and hence atomic formulae in their signatures may not be decidable. The solution given here is to develop an appropriate concept of realizability of existential assertions over such algebras, generalized to *realizability of sequents of existential assertions*. In this way, the results can be seen to hold for classical proof systems.

This investigation may give some insight into the relationship between specifiability and computability for data types such as the reals, where the atomic formulae, *i.e.*, equations between terms of type real, are not computable.

Key words and phrases: generalized computability, realizability, selection function

1 Introduction

1.1 Background: Parsons-Mints-Takeuti theorem; Attempted generalizations

We investigate a class of problems concerning the relationship between specifiability and computability for a wide class of abstract data types, modelled as many-sorted algebras A, of the following form. Given a predicate P of a certain syntactic class in the specification language $\boldsymbol{Lang}(A)$ for A, and a proof of the assertion

$$\forall \mathtt{x}\, \exists \mathtt{y}\, P(\mathtt{x}, \mathtt{y}) \tag{1.1}$$

in a suitable formal system $\mathcal{F}$ for A, can we construct, from this proof, a computable selection function for P, *i.e.*, a computable function $\mathtt{f}$ on A such that

$$\forall \mathtt{x}\, P(\mathtt{x}, \mathtt{f}(\mathtt{x})) \tag{1.2}$$

holds in A? A positive answer to this question, under sutiable conditions, will be called a *selection theorem*. (Here the notion of "computable on A" must also be explicated.)

Specifically, we want to generalize to such algebras a classical proof-theoretic result, due (independently) to Parsons [Par71, Par72], Mints [Min73], and Takeuti [Tak75, remark after Cor. 12.16], which gives a positive solution to the above problem in the case that $\mathcal{F}$ is Peano arithmetic (PA) with induction restricted to $\mathbf{\Sigma}^0_1$ formulae, P is a $\mathbf{\Sigma}^0_1$ predicate of PA, in which case a primitive recursive selection function $\mathtt{f}$ can then be found. As a corollary, a general recursive function which is provably total in PA with $\mathbf{\Sigma}^0_1$-induction is (extensionally equivalent to) a primitive recursive function.

In [TZ93] this result was generalized to predicates over many-sorted signatures Σ containing the boolean and natural sorts, with their standard operations, and abstract many-sorted Σ-algebras A. The method used was adapted from Mints's method, involving cut-reduction and an analysis of cut-reduced derivations, with restricted ($\mathbf{\Sigma^*_1}$) induction. The result used a generalization of primitive recursive schemes to many-sorted signatures and algebras. The generalization went quite smoothly, on the assumption that *equality in A was computable*, so that the atomic formulae of the first-order language over Σ were computably decidable in A.

The case that equality in A is *not computable* provides a difficulty for this generalization. In such a case, a more delicate analysis of formal derivations of assertions of the form (1.1) is required.

To clarify these issues by an example, consider the topological total algebra of reals

$$\mathcal{R} \;=\; (\mathbb{R},\, \mathbb{N},\, \mathbb{B};\; 0,\, 1,\, +,\, -,\, \times,\;\; \ldots\;\;), \tag{1.3}$$

("topological" in the sense that all the carriers have topologies in terms of which the basic operations are continuous; "total" in the sense that the basic operations are total [TZ05]). The algebra $\mathcal{R}$ contains the carrier $\mathbb{R}$ of reals with its usual topology and its ring operations, as well as the carriers $\mathbb{N}$ and $\mathbb{B}$ of naturals and booleans, with their discrete topologies and standard operations. Note that there is no division operation on $\mathbb{R}$, since

no such (total) operation can be continuous. Similarly, although there is an equality test (*i.e.*, a boolean valued equality operation) on $\mathbb{N}$, there is none on $\mathbb{R}$, since a (total) equality operation on $\mathbb{R}$ cannot be continuous.[1]

However the *specification language* $\boldsymbol{Lang}(A)$, in which the predicates P (1.1) are expressed, has, as atomic formulae, equations between terms of the same sort, for *all sorts* of A, including, *e.g.*, the sort of reals in the above example. It follows that *the atomic formulae in* $\boldsymbol{Lang}(A)$ *are not computable*.

This problem was solved in [Zuc06], by using, not just a primitive recursive *selector* for an existential statement, but a primitive recursive *realiser* for each formula, which also carries information on *which component of a disjunction* holds (as in the antecedent of the conclusion of the $\vee\mathsf{L}$ inference). However this technique only worked by restricting attention to *intuitionistic deductive systems*. Hence, the resulting selection theorem could not really be called a generalization of the Parsons-Mints-Takeuti theorem.

1.2 The present work

This problem of the restriction to intuitionistic systems has now been solved by extending the notion of realizability to *sequents* as well as formulae, as was done in [Str03]. The resulting selection theorem, in which *neither* the decidability of atomic formulae, *nor* the use of intuitionistic deductive systems, need to be assumed, is a genuine generalization of the Parsons-Mints-Takeuti theorem, and forms the main result of this paper.

This investigation may give some insight into the relationship between specifiability and computability for data types such as the reals, where the atomic formulae, *i.e.*, equations between terms of type real, are not computable.

In particular, it provides an example, in the context of verifiable specifications on such data types, of the general programme proposed by Kreisel [Kre71] of discovering "what more we know when we have proved a theorem than if we only know that it is true".

1.3 Previous work in realizability and related selection theorems

Realisability, as a technique in proof theory, goes back to [Kle45]. Since then many variants have been developed. Thorough treatments of various versions of realizability applied to Heyting arithmetic and related systems, with extensive bibliography, are given in [Tro93, Tro98].

With regard to fragments of arithmetic and related systems: apart from the pioneering work of Parsons, Mints and Takeuti mentioned above [Par71, Par71, Min73, Tak75], a number of researchers have explored selection and realizability methods for various fragments, not all assuming decidability of equality. Sieg [Sie91] described a generic Skolemisation method for subsystems of arithmetic. Buss [Bus98a] described various "witnessing methods" in fragments of arithmetic, which have been very successfully applied, especially in weak bounded arithmetics [Bus86]. Both assume decidability of equality (as in Section 5 of the present paper). Leivant [Lei94] used realizability methods for characterising poly-time

[1] One can define continuous partial division and equality operations on the reals [TZ04]; however in this paper we only consider total algebras. This is discussed further in Section 8.

functions, using Herbrand-Gödel equations with a weak second order intuitionistic logic, in which decidability of equality is not assumed (as in Section 6 of the present paper). Schlüter [Sch95] extended Leivant's result to realizability of classical sequents.

The latter technique for realising classical sequents has been used more recently in Feferman-style self-applicative systems, which form the operational core of Feferman's explicit mathematics [Fef75, Fef79]. The paper [Str03] studies a whole family of bounded applicative theories and their relation to complexity classes, whereas Cantini [Can02] gave a perspicuous characterisation of the poly-time functions by using a form of safe induction in an applicative context. The papers [Str04, Can05] contain extensions of the results in [Str03]. As with the realizability studied in Section 7 of the present paper, equality cannot be assumed to be decidable in self-applicative theories.

It should be noted that the present paper, as well as [TZ93, Zuc06], deal with a fragment (namely $\mathbf{\Sigma_1^*}$ induction), not specifically of arithmetic, but more generally, of proof systems for abstract many-sorted algebras.

1.4 Outline of this paper

Section 2 provides a short background to N-standard many-sorted signatures and algebras, *i.e.*, many-sorted signatures and algebras with the sorts of booleans and naturals, with the standard operations on these. Section 3 explains the generalization of primitive recursiveness to such signatures and algebras, and Section 4 describes the corresponding specification languages.

To provide background and context for the main results of this paper, Sections 5 and 6 summarise the two previous (restricted) generalizations of the Parsons-Mints-Takeuti theorem mentioned above: Section 5 for algebras with decidable equality, and Section 6 for intuitionistic deductive systems.

Section 7 gives the main result of this paper: the generalized selection theorem, without either of the two restrictions needed in Sections 5 and 6; *i.e.*, not assuming decidability of equality, and working in a classical deductive system. Section 8 gives some concluding remarks.

2 Many-sorted signatures and algebras

We give a short introduction to many-sorted algebras. Details may be found in any of [TZ99, TZ00, TZ04, TZ05]. Given a signature Σ with finitely many *sorts* $s, \ldots$ and *function symbols*

$$\mathsf{F} \colon u \to s, \tag{2.1}$$

where u is the product type $u = s_1 \times \cdots \times s_m$, a Σ-algebra A consists of a carrier A_s for each Σ-sort s, and a total function

$$\mathsf{F}^A \colon A^u \to A_s$$

for each Σ-function symbol as in (2.1), where $A^u = A_{s_1} \times \cdots \times A_{s_m}$. We let $s, \ldots$ range over Σ-sorts, and $u, v, w, \ldots$ over Σ-product types.

We are interested in signatures and algebras with certain properties

2.1 N-standard signatures and algebras

The signatures Σ and Σ-algebras A are said to be *N-standard* if they contain

(a) the sort bool of *booleans* and the corresponding carrier $A_{\mathsf{bool}} = \mathbb{B} = \{\mathbb{t}, \mathbb{f}\}$, together with the standard boolean and boolean-valued operations, including the conditional at all sorts, and equality at certain sorts ("equality sorts"); and also

(b) the sort nat of *natural numbers* and the corresponding carrier $A_{\mathsf{nat}} = \mathbb{N} = \{0, 1, 2, \ldots\}$, together with the standard arithmetical operations of zero, successor, equality and order on $\mathbb{N}$.

We make two assumptions on our signatures Σ and Σ-algebras A.

Assumption 1 (N-standardness). *The signatures and Σ-algebras are N-standard.*

Assumption 2 (Instantiation). *For every sort s of Σ, there is a closed term of sort s, called the default term $\boldsymbol{\delta}^s$ of that sort.*

The Instantiation Assumption will be used in the proof of the Main Lemma in Sec. 7.

Let $\boldsymbol{NStdAlg}(\Sigma)$ denote the class of N-standard algebras over Σ.

2.2 Array signatures and algebras

Array signatures Σ^* and *array algebras* A^*, are formed from N-standard signatures Σ and algebras A by adding, for each sort s, an *array sort* s^*, with corresponding carrier A^*_s consisting of all arrays or finite sequences over A_s, together with certain standard array operations. Details are given in [TZ00] and (an equivalent but simpler version) in [TZ99, TZ02].

We will generally work with array signatures and algebras, for reasons that will become clear below.

3 Computation schemes

We will present two systems of computation schemes over Σ: PR and μPR.

3.1 PR(Σ) and PR*(Σ) computation schemes

Given an N-standard signature Σ, we define PR *schemes* over Σ which generalize 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:

(i) ***Primitive Σ-functions:***

$$\mathsf{f}(x) \;=\; \mathsf{F}(x)$$

of type $u \to s$, for all the primitive Σ-function symbols $\mathsf{F}\colon u \to s$, where $\mathtt{x} : u$, *i.e.*, $\mathtt{x}$ is a tuple of variables of product type u.

(ii) ***Projection:***

$$\mathsf{f}(\mathtt{x}) \;=\; \mathtt{x}_i$$

of type $u \to s_i$, where $\mathtt{x} = (\mathtt{x}_1, \ldots, \mathtt{x}_m)$ is of type $u = s_1 \times \cdots \times s_m$.

(iii) ***Composition:***

$$\mathsf{f}(\mathtt{x}) \;=\; \mathsf{h}(\mathsf{g}_1(\mathtt{x}), \ldots, \mathsf{g}_m(\mathtt{x}))$$

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

(iv) ***Definition by cases:***

$$f(\mathtt{b}, \mathtt{x}, \mathtt{y}) \;=\; \begin{cases} \mathtt{x} & \text{if } \mathtt{b} = \mathbb{t} \\ \mathtt{y} & \text{if } \mathtt{b} = \mathbb{f} \end{cases}$$

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 $\mathsf{f} = (\mathsf{f}_1, \ldots, \mathsf{f}_m)$ with $\mathsf{f}_i : \mathsf{nat} \times u \to s_i$, such that for all $x \in A^u$ and $i = 1, \ldots, m$,

$$\begin{aligned} \mathsf{f}_i(0, \mathtt{x}) \;&=\; g_i(\mathtt{x}) \\ \mathsf{f}_i(\mathtt{z}+1, \mathtt{x}) \;&=\; \mathsf{h}_i(\mathtt{z}, \mathtt{x}, \mathsf{f}_1(\mathtt{z}, \mathtt{x}), \ldots, \mathsf{f}_m(\mathtt{z}, \mathtt{x})) \end{aligned}$$

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

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

A PR(Σ) scheme $\alpha\colon u \to s$ 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 $\boldsymbol{NStdAlg}(\Sigma)$.

A broader class of functions provides a more appropriate generalization of the notion of primitive recursiveness for our purposes, namely PR* *computability*. A function on A is PR$^*(\Sigma)$ computable if it is defined by a PR scheme over Σ^*, interpreted on A^* (*i.e.*, using starred sorts for the auxiliary functions used in its definition). Note that in the classical

setting ($A = \mathcal{N} =$ the naturals with their standard operations) this generalization is not necessary, since $\mathcal{N}^*$ can effectively be coded in $\mathcal{N}$. In general, however, this is not the case; $\mathcal{R}^*$, for example, cannot be effectively coded in $\mathcal{R}$.

We write PR(A) for the class of functions PR computable on A, etc.

3.2 μPR(Σ) and μPR* (Σ) computation schemes

The μPR *schemes* over Σ are formed by adding to the PR schemes the scheme:

(vi) ***Least number*** or μ ***operator:***

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

of type $u \to \mathsf{nat}$, where $\mathsf{g} : u \times \mathsf{nat} \to \mathsf{bool}$ is μPR. The interpretation of this is that $\mathsf{f}^A(x) \downarrow z$ if, and only if, $\mathsf{g}^A(x,y) \downarrow \mathbb{f}$ for each $y < z$ and $\mathsf{g}^A(x,z) \downarrow \mathbb{t}$.

Note that this scheme also uses the N-standardness of the algebra. Also, μPR computable functions are, in general, *partial*. The notation $f(x) \downarrow y$ means 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 generalize those given in [Kle52] for partial recursive functions over $\mathbb{N}$.

Again, a broader class turns out to be more appropriate for our purposes, namely μPR* *computability*. This is just PR* computability with μ.

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 *generalized 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.*

3.3 Comparison with imperative computational models

In [TZ00] computation on many-sorted Σ-algebras was investigated, using imperative programming models: ***While***(Σ), based on the 'while' loop construct over Σ, ***For***(Σ), based similarly on the 'for' loop, and ***While**** (Σ) and ***For**** (Σ), 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:

(1) PR(A) = ***For***(A)
(2) PR*(A) = ***For**** (A)
(3) μPR(A) = ***While***(A)
(4) μPR*(A) = ***While****(A),

in all cases, uniformly for $A \in$ ***NStdAlg***(Σ).

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

4 The language $\boldsymbol{Lang}^*(\Sigma)$; $\boldsymbol{\Sigma_1^*}$ formulae; the system $\boldsymbol{\Sigma_1^*}$-Ind

4.1 The language $\boldsymbol{Lang}^*(\Sigma)$

We let $\boldsymbol{Lang}(\Sigma)$ denote the first order language over Σ, and let $\boldsymbol{Lang}^*(\Sigma) = \boldsymbol{Lang}(\Sigma^*)$, the first order language over Σ^*. The *atomic formulae* of $\boldsymbol{Lang}(\Sigma)$ are equations between terms of the same sort, *for all Σ-sorts* (not just equality sorts). Similarly, $\boldsymbol{Lang}^*(\Sigma) = \boldsymbol{Lang}(\Sigma^*)$ is the first order language over Σ^*, with equality at all Σ^*-sorts.

Notation. (1) We use $\mathtt{x}$, $\mathtt{y}$, $\mathtt{z}$, ... for variables or tuples of variables, $\mathtt{x}^*$ ... for starred (or array) variables or tuples of variables, $\mathtt{k}, \ldots$ for variables of sort nat, and $t, t', \ldots$ for Σ^*-terms or tuples of terms. We write $t : s$ to indicate that t is a term of sort s, and $t : u$ that t is a tuple of terms of product type u.

(2) We define application of function tuples to argument tuples in the obvious way, *i.e.*, if $\mathsf{f}\colon u \to v$ is a tuple of function symbols $(\mathsf{f}_1, \ldots, \mathsf{f}_m)$ where $\mathsf{f}_i\colon u \to s_i$ $(i = 1, \ldots, m)$ with $v = s_1 \times \cdots \times s_m$, and $\mathtt{x} : u$, then $\mathsf{f}(\mathtt{x}) \equiv_{df} (\mathsf{f}_1(\mathtt{x}), \ldots, \mathsf{f}_m(\mathtt{x}))$.

Our proof system is based on the *classical sequent calculus* [Gen69, Tak75] with sequents

$$\Gamma \longmapsto \Delta, \tag{4.1}$$

where Γ and Δ are finite sequences of formulae of $\boldsymbol{Lang}^*(\Sigma)$, with the informal meaning: the conjunction of the antecedent Γ implies the disjunction of the succedent Δ. (Unlike [Gen69, Tak75], however, we will place our principle formulae on the "inside" of the sequents, to simplify the notation in the later sections.)

We are interested in a certain sublanguage of $\boldsymbol{Lang}^*(\Sigma)$, namely the class of $\boldsymbol{\Sigma_1^*}$ formulae over Σ, which we now define.

4.2 Subclasses of $\boldsymbol{Lang}^*(\Sigma)$.

(a) ***BU quantifiers, equations and sequents.***

(i) A *BU (bounded universal) quantifier* is a quantifier of the form '$\forall \mathtt{k} < t$', where $\mathtt{k} : \mathsf{nat}$ and $t : \mathsf{nat}$. (The most elegant approach is to think of this as a primitive construct, with its own introduction rule: see below.)

(ii) A *BU equation* is formed by prefixing an equation by a string of 0 or more bounded universal quantifiers.

(iii) A *conditional BU equation* is a formula of the form

$$Q_1 \wedge \ldots \wedge Q_n \to P \tag{4.2}$$

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

(iv) A *BU equational sequent* is a sequent of the form

$$Q_1, \ldots, Q_n \;\longmapsto\; P \tag{4.3}$$

where the Q_i and P are BU equations. This sequent *corresponds to* the conditional BU equation (4.2).

(*b*) ***Elementary formulae***.
A formula of $\boldsymbol{Lang}^*(\Sigma)$ is *elementary* if it is formed from Σ^*-equations by applying *conjunctions*, *disjunctions*, and *BU quantification* (in any order).

(*c*) $\boldsymbol{\Sigma_1^*}$ ***formulae***[2].
A formula is $\boldsymbol{\Sigma_1^*}$ if it is formed from Σ^*-equations by applying *conjunctions*, *disjunctions*, *BU quantification* and also *existential Σ^*-quantification*, *i.e.*, unbounded existential quantification over any sort in Σ^* (in any order).

(*d*) ***Prenex $\boldsymbol{\Sigma_1^*}$ formulae.***
A formula is in *prenex* $\boldsymbol{\Sigma_1^*}$ if it is formed from an elementary formula by applying (0 or more) *existential Σ^*-quantifications*, only.

Lemma 1 (Prenex form of a $\boldsymbol{\Sigma_1^*}$ formula). *Every $\boldsymbol{\Sigma_1^*}$ formula is effectively equivalent to a prenex $\boldsymbol{\Sigma_1^*}$ formula, provably in the intuitionistic system $\boldsymbol{\Sigma_1^*}$-$\mathsf{Ind_i}$ (defined in §4.3 below).*

The construction of the prenex form is by structural induction on the formula. In the case of permuting an '$\exists$' with a BU quantifier, the existentially quantified variable changes to a starred sort (if it is not already starred):

$$\forall \mathtt{k} < t\, \exists \mathtt{x}\, P(\mathtt{k}, \mathtt{x}) \quad \longmapsto \quad \exists \mathtt{x}^*\, \forall \mathtt{k} < t\, P(\mathtt{k}, \mathtt{x}^*[\mathtt{k}]).$$

Some details of the intuitionistic derivability of this sequent are given in [TZ93].

Lemma 2. *If P is an elementary formula all of whose variables are of equality sort, then the predicate defined by P is* PR* *computable.*

Let T be a set of formulae in $\boldsymbol{Lang}^*$, which we can think of as *axioms* for a class of Σ^*-algebras. We make the following assumption about T.

Conditional BU Axiomatisation Assumption. The axiomatisation T consists of conditional BU Σ^*-equations.

Note that this is a stricter condition than conditional $\boldsymbol{\Sigma_1^*}$ formulae, since it excludes disjunctions and existential quantification. However, this assumption is not unduly restrictive, as it includes axiomatisations by *conditional equations*, and (hence) *Horn formulae*, which are central to the theory of logic programming and abstract data types [MT92].

We will define a sequent calculus $\boldsymbol{\Sigma_1^*}$-$\mathsf{Ind}(\Sigma, T)$ with the axioms T as extra initial sequents.

[2]The notation may be a bit confusing: Σ^* refers to a signature with array sorts, whereas $\boldsymbol{\Sigma_1^*}$ refers to a particular syntactic class of formulae over Σ^*.

4.3 The classical sequent calculus $\mathbf{\Sigma_1^*}$-$\mathsf{Ind}(\Sigma, T)$

This system has the following inference rules: rules for the first order predicate calculus with equality over the signature Σ^*, including *cut* as in [Gen69, Tak75]; the $\mathbf{\Sigma_1^*}$ induction rule

$$\frac{\Gamma,\ P(\mathsf{a}) \ \longmapsto\ P(\mathsf{Sa}),\ \Delta}{\Gamma,\ P(0) \ \longmapsto\ P(t),\ \Delta}. \tag{4.4}$$

where the induction formula $P(\mathsf{a})$ is $\mathbf{\Sigma_1^*}$, and the induction variable $\mathtt{a}\colon \mathsf{nat}$ does not occur in Γ, Δ or $P(0)$; and rules for the BU quantifier:

$$\forall_b L: \frac{\Gamma \ \longmapsto\ t_0 < t,\ \Delta \qquad \Gamma,\ Q(t_0) \ \longmapsto\ \Delta,}{\Gamma,\ \forall \mathtt{k} < t\ Q(\mathtt{k}) \ \longmapsto\ \Delta} \qquad \forall_b R: \frac{\Gamma,\ \mathtt{a} < t \ \longmapsto\ P(\mathtt{a}),\ \Delta}{\Gamma \ \longmapsto\ \forall \mathtt{k} < t\, P(\mathtt{k}),\ \Delta}$$

where t_0 and t are terms of sort nat, and $\mathtt{a} : \mathsf{nat}$ is the '*eigenvariable*' of the inference $\forall_b R$, which does not occur in the conclusion of that inference. (We could also add two rules for the bounded existential quantifier, dual to the above, although this quantifier is not really needed in the subsequent development.)

The axioms (initial sequents) are the closures under substitution of the following: the Σ^*-equality axioms; the standard axioms for bool, including

$$\longmapsto\ (\mathtt{x}^{\mathsf{bool}} = \mathsf{true})\ \vee\ (\mathtt{x}^{\mathsf{bool}} = \mathsf{false}), \tag{4.5a}$$

$$\mathsf{true} = \mathsf{false} \ \longmapsto\ t_1 = t_2 \tag{4.5b}$$

for arbitrary terms t_1, t_2 of the same sort; the axioms for zero and successor on nat:

$$\mathsf{S}\,\mathtt{m} = \mathsf{S}\,\mathtt{n} \ \longmapsto\ \mathtt{m} = \mathtt{n},$$
$$\mathsf{S}\,\mathtt{n} = 0 \ \longmapsto\ t_1 = t_2$$

for nat variables $\mathtt{m}$, $\mathtt{n}$ and arbitrary terms t_1, t_2 of the same sort; the primitive recursive defining equations for '$<$' on nat (which is used in the BU quantifier rules and array axioms), and (optionally) symbols and defining equations for other primitive recursive functions on nat; a certain set of conditional BU axioms for arrays[3], including the BU equational sequent for array equality:

$$\mathsf{Lgth}(\mathtt{a}_1^*) = \mathsf{Lgth}(\mathtt{a}_2^*) \wedge \forall \mathtt{z} < \mathsf{Lgth}(\mathtt{a}_1^*)\big(\mathtt{a}_1^*[\mathtt{z}] = \mathtt{a}_2^*[\mathtt{z}]\big) \ \longmapsto\ \mathtt{a}_1^* = \mathtt{a}_2^* \tag{4.6}$$

and, finally, the axioms T in sequent form (*cf.* §4.2(a)(iv)).

Remarks (Initial sequents). (1) It follows from the Axiomatisation Assumption that the initial sequents of the calculus $\mathbf{\Sigma_1^*}$-$\mathsf{Ind}(\Sigma, T)$ are all $\mathbf{\Sigma_1^*}$. In fact, they are all *BU equational* (except for (4.5a), which is a disjunction of equations). This is important for the proof of the Main Lemma in Sections 5, 6 and 7.

(2) The initial sequents were defined so as to be closed under substitution. This is to facilitate the proof of the cut reduction lemma (§5.1).

[3] listed in [TZ02, §3.2]

Now let $\mathbb{K} \subseteq \boldsymbol{NStdAlg}(\Sigma)$, and let T be a set of formulae in $\boldsymbol{Lang}^*$ such that $\mathbb{K} \models T$. (We could suppose that T is a "complete N-standard axiomatisation" for $\mathbb{K}$, *i.e.*, that $\mathbb{K}$ is the class of all N-standard Σ-structures satisfying T, although this is unnecessary for the subsequent development.) The following soundness result then clearly holds:

Lemma 2 (Soundness of $\boldsymbol{\Sigma_1^*}$-Ind). $\quad \boldsymbol{\Sigma_1^*}\text{-}\mathsf{Ind}(\Sigma, T) \vdash P \implies \mathbb{K}^* \models P.$

4.4 The intuitionistic sequent calculus $\boldsymbol{\Sigma_1^*}$-$\mathsf{Ind}_\mathsf{i}(\Sigma, T)$

This consists of intuitionistic sequents of the form (4.1), where Δ consists of exactly one formula. The inference rules have their intuitionistic form, as described in [Gen69, Tak75]. In particular, the *intuitionistic induction rule* has the form (4.4) with Δ empty. Note also that by Assumption 3, the axioms T have the form (4.3) of intuitionistic sequents.

Since $\boldsymbol{\Sigma_1^*}\text{-}\mathsf{Ind}_\mathsf{i}$ is a subsystem of $\boldsymbol{\Sigma_1^*}\text{-}\mathsf{Ind}$, the soundness lemma (Lemma 2) obviously still holds for $\boldsymbol{\Sigma_1^*}\text{-}\mathsf{Ind}_\mathsf{i}$.

4.5 Equational specifications of $\mathbf{PR}^{(*)}$ functions

For any $\mathrm{PR}(\Sigma)$ scheme α, we can construct a *equational specification*, *i.e.*, a finite set E_α of "specifying equations" for the function f_α^A, defined by α on all $A \in \boldsymbol{NStdAlg}(\Sigma)$, as well as for the auxiliary functions g_α used in the definition of α. The set E_α consists of equations in an *expanded signature* $\Sigma_\alpha = \Sigma \cup \{\mathsf{g}_\alpha, \mathsf{f}_\alpha\}$. It is defined by structural induction on α.

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

Although we do not use the following in this paper, we mention that for $\mu\mathrm{PR}^*$ schemes α, we can similarly construct a *conditional BU equational specification* in an expanded signature $\Sigma_\alpha^* = \Sigma^* \cup \{\mathsf{g}_\alpha, \mathsf{f}_\alpha\}$, which specifies f_α^A on all N-standard Σ-algebras A in which f_α^A is total. Note that *conditional BU* equations are needed for the specification of the μ operator.

Details of the above can be found in [TZ02].

4.6 $\boldsymbol{\Sigma_1^*}$ computation predicates; Provable totality of schemes

We present another specification system for schemes, using $\boldsymbol{\Sigma_1^*}$ predicates, but not expanded signatures.

With each $\mu\mathrm{PR}^*(\Sigma)$ scheme $\alpha\colon u \to s$, we can effectively associate a $\boldsymbol{\Sigma_1^*}(\Sigma)$ formula $P_\alpha(\mathtt{x}, \mathtt{y})$, the *computation predicate for* α, where $\mathtt{x}\colon u$ and $\mathtt{y}\colon s$, which represents the graph of the function defined by α, *i.e.*, for all $A \in \boldsymbol{NStdAlg}(\Sigma)$, and for all $a \in A^u$ and $b \in A_s$,

$$A \models P_\alpha[a, b] \iff \alpha^A(a) \downarrow b.$$

The construction of P_α is by structural induction on α. Details can be found in [TZ93].

Note that even if the scheme α is defined over Σ only, *i.e.*, $\mu\mathrm{PR}$ or even PR, the definition of P_α generally involves existential quantification over *starred sorts*.

A scheme α is said to be *provably total in* $\mathbf{\Sigma_1^*}\text{-}\mathsf{Ind}(\Sigma, T)$ iff

$$\mathbf{\Sigma_1^*}\text{-}\mathsf{Ind}(\Sigma, T) \ \vdash \ \forall \mathsf{x}\, \exists \mathsf{y} P_\alpha(\mathsf{x}, \mathsf{y}).$$

Lemma (Totality for PR* schemes).
If α is a PR* *scheme, then α is provably total in* $\mathbf{\Sigma_1^*}\text{-}\mathsf{Ind}_\mathsf{i}(\Sigma)$.

The required derivation is constructed by structural induction on α. Details can be found in [TZ93].

5 Selection theorem for algebras with computable equality

5.1 Statements of main results

The central result of this paper is formulated with reference to a class $\mathbb{K}$ of N-standard Σ-algebras and an axiomatisation T of $\mathbb{K}$.

Theorem 1 (Selection Theorem). *Suppose* $\mathbb{K} \models T$ *where* $\mathbb{K} \subseteq \boldsymbol{NStdAlg}(\Sigma)$*, and* T *consists of conditional BU* Σ^**-equations. If*

$$\mathbf{\Sigma_1^*}\text{-}\mathsf{Ind}(\Sigma, T) \ \vdash \ \exists \mathtt{y} P(\mathtt{x}, \mathtt{y})$$

where $P(\mathtt{x}, \mathtt{y})$ *is an elementary formula, with free variables* $\mathtt{x}\colon u$ *and* $\mathtt{y}\colon v$*, then there is a* PR* *scheme tuple* $\alpha\colon u \to v$ *such that*

$$\text{for all } A \in \mathbb{K}, \text{ and all } x \in A^u, \quad A \ \models \ P[x, \mathsf{f}_\alpha^A(x)]. \tag{5.1}$$

The function (tuple) f_α^A is called a *selecting function, realising function, Skolem function* or *witnessing function* for $\mathtt{y}$ in P.

As a corollary, we have a kind of converse to the Totality Lemma in §4.5.

Corollary. *Suppose* $\mathbb{K} \models T$ *where* $\mathbb{K} \subseteq \boldsymbol{NStdAlg}(\Sigma)$ *and* T *consists of conditional BU* Σ^**-equations. If a* μPR* *scheme* α *is provably total in* $\mathbf{\Sigma_1^*}\text{-}\mathsf{Ind}(\Sigma, T)$*, then* α *is extensionally* PR* *on* $\mathbb{K}$*, i.e., there is a* PR* *scheme* β *such that* $\mathsf{f}_\alpha^A = \mathsf{f}_\beta^A$ *for all* $A \in \mathbb{K}$.

A stronger version of Theorem 1 involves replacing (5.1) by a *provability* condition:

Theorem 2 (Provable Selection Theorem). *Suppose* T *consists of conditional BU* Σ^**-equations. If*

$$\mathbf{\Sigma_1^*}\text{-}\mathsf{Ind}(\Sigma, T) \ \vdash \ \exists \mathtt{y} P(\mathtt{x}, \mathtt{y})$$

where $P(\mathtt{x}, \mathtt{y})$ *is an elementary formula, with free variables* $\mathtt{x}\colon u$ *and* $\mathtt{y}\colon v$*, then there is a* PR* *scheme tuple* $\alpha\colon u \to s$ *such that*

$$\mathbf{\Sigma_1^*}\text{-}\mathsf{Ind}(\Sigma_\alpha^*, T + E_\alpha) \ \vdash \ P(\mathtt{x}, \mathsf{f}_\alpha(\mathtt{x}))$$

where Σ_α^* *is the extension of* Σ^* *with symbols for the functions* $\mathsf{f}_\alpha\colon u \to v$ *defined by the scheme tuple* α*, together with their auxiliary functions, and* E_α *is the equational specification for these functions given in §4.4.*

Theorem 1 is an immediate consequence of Theorem 2. Theorem 2, in turn, follows immediately from a more general result. We first need some definitions and notation.

Definitions ($\mathbf{\Sigma_1^*}$ sequent and derivation).

(1) A sequent is called $\mathbf{\Sigma_1^*}$ if all its formulae are $\mathbf{\Sigma_1^*}$.

(2) A derivation is called $\mathbf{\Sigma_1^*}$ if all its sequents are $\mathbf{\Sigma_1^*}$.

Definitions and notation (Prenex form of a sequent).
In this section (only) we use the following notation.

(3) For any $\mathbf{\Sigma_1^*}$ formula $P(\mathtt{x})$ containing (only) the variables $\mathtt{x}$ free, we write its prenex form (§4.1, Lemma 1) as $\exists \mathtt{y} P^0(\mathtt{x}, \mathtt{y})$, with P^0 elementary.

(4) Given a $\mathbf{\Sigma_1^*}$ sequent

$$Q_1, \ldots, Q_m \;\longmapsto\; P_1, \ldots, P_n, \tag{5.2}$$

its *prenex form* is the corresponding sequent of prenex forms of the formulae:

$$\exists \mathtt{z}_1 Q_1^0(\mathtt{x}, \mathtt{z}_1),\ \ldots,\ \exists \mathtt{z}_m Q_m^0(\mathtt{x}, \mathtt{z}_m) \;\longmapsto\; \exists \mathtt{y}_1 P_1^0(\mathtt{x}, \mathtt{y}_1),\ \cdots,\ \exists \mathtt{y}_n P_n^0(\mathtt{x}, \mathtt{y}_n) \tag{5.3}$$

where $\mathtt{x}$ contains all free variables of the sequent.

Main Lemma. *Suppose the* $\mathbf{\Sigma_1^*}$ *sequent* (5.2) *is provable in* $\mathbf{\Sigma_1^*}$-$\mathsf{Ind}(\Sigma, T)$. *Let its prenex form be as in* (5.3). *Then we can construct tuples of* $\mathrm{PR}^*(\Sigma)$ *schemes* $\alpha_1, \ldots, \alpha_n$ *such that*

$$Q_1^0(\mathtt{x}, \mathtt{z}_1),\ \ldots,\ Q_m^0(\mathtt{x}, \mathtt{z}_m) \;\longmapsto\; P_1^0(\mathtt{x}, \mathsf{f}_{\alpha_1}(\mathtt{x}, \mathtt{z})),\ \cdots,\ P_n^0(\mathtt{x}, \mathsf{f}_{\alpha_n}(\mathtt{x}, \mathtt{z})) \tag{5.4}$$

(where $\mathtt{z} \equiv \mathtt{z}_1, \ldots, \mathtt{z}_m$*) is provable in* $\mathbf{\Sigma_1^*}$-$\mathsf{Ind}(\Sigma^*_{\alpha_1,\ldots,\alpha_n}, T + E_{\alpha_1,\ldots,\alpha_n})$, *where* $E_{\alpha_1,\ldots,\alpha_n}$ *is the combined equational specification for the functions* $\mathsf{f}_{\alpha_1}, \ldots, \mathsf{f}_{\alpha_n}$ *in the signature* $\Sigma^*_{\alpha_1,\ldots,\alpha_n}$.

In order to prove the Main Lemma, we must first prove a *cut reduction lemma*.

Cut reduction lemma. Every derivation $\mathcal{D}$ in $\mathbf{\Sigma_1^*}$-Ind, with $\mathbf{\Sigma_1^*}$ initial sequents, can be transformed into a derivation $\mathcal{D}'$ of the same end-sequent containing only $\mathbf{\Sigma_1^*}$ cuts. Moreover, if the end-sequent is $\mathbf{\Sigma_1^*}$ then so is the whole derivation.

The proof of this lemma proceeds by a technique similar to that in the proof of Gentzen's *Hauptsatz* (see [Gen69, III, §3] or [Tak75, §5]). Details are given in [TZ93].

5.2 Proof of main lemma

By the Cut Reduction Lemma and the Remark on initial sequents in §4.2, we can assume we have a $\mathbf{\Sigma_1^*}$ derivation of (5.2).

There are different cases according to the last inference. It is given in some detail in [TZ93]. We cover a few cases that most concern us.

The result holds trivially for *initial sequents*, by the Remark on initial sequents in §4.2.

A PR^* selection function for $\mathbf{\Sigma_1^*}$ *induction* can be defined by the scheme for primitive recursion.

Consider now $\mathsf{Contr{:}R}$. Rewriting the premiss and conclusion in prenex form, we have:

$$\frac{\ldots,\ \exists \mathtt{z}_j Q_j^0(\mathtt{x}, \mathtt{z}_j),\ \ldots \;\longmapsto\; \exists \mathtt{y} P^0(\mathtt{x}, \mathtt{y}),\ \exists \mathtt{y} P^0 \mathtt{x}, \mathtt{y}),\ \ldots}{\ldots,\ \exists \mathtt{z}_j Q_j^0(\mathtt{x}, \mathtt{z}_j),\ \ldots \;\longmapsto\; \exists \mathtt{y} P^0(\mathtt{x}, \mathtt{y}), \ldots}\ .$$

By induction hypothesis there are PR* functions $\mathsf{f}_1, \mathsf{f}_2$ such that

$$\ldots,\ Q^0_j(\mathtt{x}, \mathtt{z}_j),\ \ldots\ \longmapsto\ \ldots,\ P^0(\mathtt{x}, \mathsf{f}_1(\mathtt{x}, \mathtt{z})),\ P^0(\mathtt{x}, \mathsf{f}_2(\mathtt{x}, \mathtt{z}))$$

is provable. So define the vector of PR functions

$$\mathsf{f}(\mathtt{x}, \mathtt{z})\ =\ \begin{cases} \mathsf{f}_1(\mathtt{x}, \mathtt{z}) & \text{if } P^0(\mathtt{x}, \mathsf{f}_1(\mathtt{x}, \mathtt{z})) \\ \mathsf{f}_2(\mathtt{x}, \mathtt{z}) & \text{otherwise} \end{cases} \tag{5.5}$$

using *definition by cases*.

Then f is a selection function for $\exists \mathtt{y} P^0$ in the conclusion.

Note that for (5.5) to define a PR* function, we need *primitive recursive decidability of elementary formulae* such as P^0.

A similar situation arises with the rules $\wedge\mathsf{R}$ and $\vee\mathsf{L}$, because of the implicit contraction of the (non-principal) formulae in the succedent. Consider, for example, the rule $\vee\mathsf{L}$:

$$\frac{\ldots,\ Q_1\ \longmapsto\ P, \ldots \qquad \ldots,\ Q_2\ \longmapsto\ P, \ldots}{\ldots,\ Q_1 \vee Q_2\ \longmapsto\ P, \ldots}. \tag{5.6}$$

Rewriting the premisses and conclusion in prenex form, we have:

$$\frac{\ldots,\ \exists \mathtt{z}_1 Q^0_1(\mathtt{x}, \mathtt{z}_1),\ \longmapsto\ \exists \mathtt{y} P^0(\mathtt{x}, \mathtt{y}),\ \ldots \qquad \ldots,\ \exists \mathtt{z}_2 Q^0_2(\mathtt{x}, \mathtt{z}_2),\ \longmapsto\ \exists \mathtt{y} P^0(\mathtt{x}, \mathtt{y}),\ \ldots}{\ldots,\ \exists \mathtt{z}_1 \mathtt{z}_2 (Q^0_1(\mathtt{x}, \mathtt{z}_1)\ \vee\ Q^0_2(\mathtt{x}, \mathtt{z}_2)),\ \longmapsto\ \exists \mathtt{y} P^0(\mathtt{x}, \mathtt{y}), \ldots}. \tag{5.7}$$

By induction hypothesis there are PR* functions $\mathsf{f}_1, \mathsf{f}_2$ such that

$$\begin{aligned} \ldots,\ Q^0_1(\mathtt{x}, \mathtt{z}_1) \quad &\longmapsto\ P^0(\mathtt{x}, \mathsf{f}_1(\mathtt{x}, \mathtt{z})),\ \ldots \\ \ldots,\ Q^0_2(\mathtt{x}, \mathtt{z}_2), \quad &\longmapsto\ P^0(\mathtt{x}, \mathsf{f}_2(\mathtt{x}, \mathtt{z})),\ \ldots \end{aligned}$$

are provable. As a selector for $\exists \mathtt{y} P^0$ in the the conclusion of (5.7), we can then define

$$\mathsf{f}(\mathtt{x}, \mathtt{z})\ =\ \begin{cases} \mathsf{f}_1(\mathtt{x}, \mathtt{z}) & \text{if } Q^0_1(\mathtt{x}, \mathtt{z}_1) \\ \mathsf{f}_2(\mathtt{x}, \mathtt{z}) & \text{otherwise} \end{cases}$$

(and similarly for the other formulae in the consequent), assuming, again, that we have PR* decidability of elementary formulae.

This is guaranteed by the assumption:

Computable Equality Assumption. All sorts of Σ have PR* computable equality.

Lemma. *Under the Computable Equality Assumption, the predicate defined by an elementary formula is* PR* *computable.*

5.3 Conclusion

Thus the Main Lemma, and hence the Selection Theorem, follow from the Computable Equality Assumption.[4]

However, *many important algebras do not have decidable equality!*

Example. Consider the topological total algebra of reals

$$\mathcal{R} \;=\; (\mathbb{R},\, \mathbb{N},\, \mathbb{B};\; 0,\, 1,\, +,\, -,\, \times,\;\; \ldots\;\;),$$

"topological" in the sense that all the carriers have topologies in terms of which the basic operations are continuous; "total" in the sense that the basic operations are total [TZ05]. $\mathcal{R}$ containing the carrier $\mathbb{R}$ of reals with its usual topology and its ring operations, as well as the carriers $\mathbb{N}$ and $\mathbb{B}$ of naturals and booleans, with their discrete topologies and standard operations.

Although there is an equality test on $\mathbb{N}$, there is none on $\mathbb{R}$, since a (total) equality operation on $\mathbb{R}$ cannot be continuous.

However the specification language $\boldsymbol{Lang}(\mathcal{R})$ has, as atomic formulae, equations between terms of the same sort, for *all sorts*, including real. Hence the atomic formulae in $\boldsymbol{Lang}(\mathcal{R})$ are not PR*-computable.

Thus we want to find conditions for the Selection Theorem which do not need the Computable Equality Assumption. We turn to this in the next two sections.

[4]This assumption was used, but its necessity was unfortunately not emphasised, in [TZ93].

6 Selection theorem for algebras with intuitionistic proof systems

6.1 Realisability

We are looking for a way to prove the Selection Theorem without assuming PR decidablity of elementary formulae, or (equivalently) of equality at all sorts.

The solution we take (for now), following [Zuc06], is to use, not just a PR *selector* for an existential statement, but a PR *realiser* for each formula, which also carries information on *which component of a disjunction* holds (as in the antecedent of the conclusion of (5.6) or (5.7)). It will turn out we also have to restrict our attention to intuitionistic systems.

We therefore define a *realizability* relation between term tuples and $\mathbf{\Sigma_1^*}$ formulae. First we define

Definition 1 (Type of a $\mathbf{\Sigma_1^*}$ formula). The *type* $\boldsymbol{tp}(P)$ of a $\mathbf{\Sigma_1^*}$ formula P is a particular Σ^*-product type. It is defined by structural induction on P.

(i) $\boldsymbol{tp}(t_1 = t_2) \;=\; \mathsf{bool}$

(ii) $\boldsymbol{tp}(P_1 \wedge P_2) \;=\; \boldsymbol{tp}(P_1) \times \boldsymbol{tp}(P_2)$

(iii) $\boldsymbol{tp}(P_1 \vee P_2) \;=\; \mathsf{bool} \times \boldsymbol{tp}(P_1) \times \boldsymbol{tp}(P_2)$

(iv) $\boldsymbol{tp}(\forall \mathtt{k} < t\, P) \;=\; \boldsymbol{tp}(P)^*$

where, for any Σ^*-product type u, u^* is the corresponding component-wise starred type; thus, if (say) $u = s_1 \times s_2 \times s_3^* \times s_4^* \times s_5$ then $u^* = s_1^* \times s_2^* \times s_3^{**} \times s_4^{**} \times s_5^*$.

(v) $\boldsymbol{tp}(\exists \mathtt{y}^s P) \;=\; s \times \boldsymbol{tp}(P)$ where s is any Σ^*-sort.

Remarks. (1) The base case, $\boldsymbol{tp}(t_1 = t_2)$, could really be defined to be any Σ-sort.

(2) The doubly starred sorts s^{**} which appear in clause (iv) are not actually present in the signature Σ^*; the doubly indexed (two-dimensional) arrays which they represent are actually effectively coded by one-dimensional arrays in a well-known way.

The central concept of this section is a *realizability* relation between term tuples of a particular Σ^*-product type, and $\mathbf{\Sigma_1^*}$ formulae of the same type.

Definition 2 (Realisability of $\mathbf{\Sigma_1^*}$ formulae). Let t be a Σ^*-term tuple, and P a $\mathbf{\Sigma_1^*}$ formula, both of the same product type. We define the expression '$t \rhd P$' ("t realises P") to be a $\mathbf{\Sigma_1^*}$ formula, by structural induction on P:

(i) $t \rhd (t_1 = t_2) \;\equiv\; t_1 = t_2$.

(ii) $\langle t_1, t_2 \rangle \rhd (P_1 \wedge P_2) \;\equiv\; (t_1 \rhd P_1) \wedge (t_2 \rhd P_2)$.

(iii) $\langle t_0, t_1, t_2 \rangle \rhd (P_1 \vee P_2) \;\equiv\; (t_0 = \mathsf{true} \wedge t_1 \rhd P_1) \vee (t_0 = \mathsf{false} \wedge t_2 \rhd P_2)$.

(iv) $t^* \rhd (\forall \mathtt{z} < t_0\, P) \;\equiv\; \forall \mathtt{z} < t_0 (t^*[\mathtt{z}] \rhd P)$.

(v) $\langle t_0, t \rangle \rhd (\exists \mathtt{y} P) \;\equiv\; t \rhd P\langle \mathtt{y}/t_0 \rangle$

Remarks. (3) If P is a formula built up from *equations* using *conjunction* and *BU quantification* only, then $t \rhd P$ is identical to P (by a simple induction on P). In particular,

the realizability of a *BU equation* P is the same as P.

(4) However, in cases (*iii*) and (v), the realising tuple contains extra information: it includes a "witness" to the truth of the disjunction or existential quantification respectively.

The above two remarks together imply that for a $\mathbf{\Sigma_1^*}$ formula P, realizability of P implies P. This is stated precisely in the following

Lemma. *For any* $\mathbf{\Sigma_1^*}$ *formula* P *and term tuple* t *of the same type, the sequent*

$$t \rhd P \;\longmapsto\; P$$

is provable in intuitionistic predicate logic.

As a sort of converse, we have the Selection Theorem (Theorems 1 and 2 of Section 5, with '$\mathbf{\Sigma_1^*}$-Ind' replaced by the intuitionistic system '$\mathbf{\Sigma_1^*}$-$\mathsf{Ind_i}$' throughout.) The Main Lemma, from which this immediately follows, asserts the existence of a realiser for the succedent formula of a $\mathbf{\Sigma_1^*}$ sequent, which is PR not just in the free variables of the sequent, but also in realisers of the antecedent formulae.

Main Lemma. Suppose the $\mathbf{\Sigma_1^*}$ sequent

$$Q_1, \ldots, Q_m \;\longmapsto\; P$$

is provable in $\mathbf{\Sigma_1^*}$-$\mathsf{Ind_i}(\Sigma, T)$. Let $Q_1, \ldots, Q_m, P$ have types $v_1, \ldots, v_m, v$ respectively, and $\boldsymbol{var}(Q_1, \ldots, Q_m, P) \subseteq \mathtt{x} : u$. Let $\mathtt{z}_1, \ldots, \mathtt{z}_m$ be tuples of variables, pairwise disjoint and disjoint from $\mathtt{x}$, with $\mathtt{z}_i \colon v_i$ for $i = 1, \ldots, m$. Then for some tuple of PR schemes $\alpha \colon u \times v_1 \times \cdots \times v_m \;\to\; v$,

$$\mathtt{z}_1 \rhd Q_1,\ \ldots,\ \mathtt{z}_m \rhd Q_m \;\longmapsto\; \mathsf{f}_\alpha(\mathtt{x}, \mathtt{z}_1, \ldots, \mathtt{z}_m) \rhd P \tag{6.1}$$

is provable in $\mathbf{\Sigma_1^*}$-$\mathsf{Ind_i}(\Sigma^*_\alpha, \mathrm{T}{+}E_\alpha)$, where Σ^*_α is the extension of Σ^* with symbols for the function tuple together with their auxiliary functions, and E_α is the equational specification for these functions.

6.2 Proof of the Main Lemma

The proof is, again, by induction on the length of a $\mathbf{\Sigma_1^*}$ derivation of (6.1) Note, in this connection, that the Cut reduction lemma also applies to the intuitionistic system.

Note also that in this proof, using realizability, we do not need to transform the sequents to prenex form (as in Section 5).

Again, there are cases according to the last inference.

We do not give a thorough proof of the Main Lemma, since such a proof is given in the next section for a stronger result. For now, we only want to consider the three inferences which (explicitly or implicitly) use contraction in the succedent in the classical case and hence needed decidability of equality, namely $\mathsf{Contr}{:}\mathsf{R}$, $\wedge\mathsf{R}$ and $\vee\mathsf{L}$, (see §5.2). First, $\mathsf{Contr}{:}\mathsf{R}$

is not part of the intuitionistic system. Secondly, $\wedge\mathsf{R}$ is no longer a problem, since the (implicit) contractions here apply only to *non-principal formulae* in the succedent, which do not exist in the intuitionistic system. That leaves $\vee\mathsf{L}$:

$$\frac{\Gamma,\, Q_1 \;\longmapsto\; P \qquad \Gamma,\, Q_2 \;\longmapsto\; P}{\Gamma,\, Q_1 \vee Q_2 \;\longmapsto\; P}$$

with all the free variables in the conclusion included in $\mathtt{x}$. By induction hypotheses there are PR* schemes α_1, α_2 such that

$$\begin{array}{rcl} \mathtt{z} \rhd \Gamma,\, \mathtt{z}_1 \rhd Q_1 & \longmapsto & \mathsf{f}_{\alpha_1}(\mathtt{x},\mathtt{z},\mathtt{z}_1) \rhd P \\ \mathtt{z} \rhd \Gamma,\, \mathtt{z}_2 \rhd Q_2 & \longmapsto & \mathsf{f}_{\alpha_2}(\mathtt{x},\mathtt{z},\mathtt{z}_2) \rhd P \end{array} \tag{6.2}$$

are provable. Define a PR* scheme tuple β such that (with $\mathtt{z}_0\colon \mathsf{bool}$, and the other variables as in (6.2))

$$\mathsf{f}_\beta(\mathtt{x},\mathtt{z},\mathtt{z}_0,\mathtt{z}_1,\mathtt{z}_2) \;=\; \begin{cases} \mathsf{f}_{\alpha_1}(\mathtt{x},\mathtt{z},\mathtt{z}_1) & \text{if } \mathtt{z}_0 = \mathsf{true} \\ \mathsf{f}_{\alpha_2}(\mathtt{x},\mathtt{z},\mathtt{z}_2) & \text{otherwise.} \end{cases}$$

Then

$$\mathtt{z} \rhd \Gamma,\; (\mathtt{z}_0,\mathtt{z}_1,\mathtt{z}_2) \rhd Q_1 \vee Q_2 \;\longmapsto\; \mathsf{f}_\beta(\mathtt{x},\mathtt{z},\mathtt{z}_0,\mathtt{z}_1,\mathtt{z}_2) \rhd P. \tag{6.3}$$

Remark: Notice here the use of the realizability property for disjunctions (see Remark (4) in §6.1) to decide effectively which component of the disjunction $Q_1 \vee Q_2$ holds. (Remember that the elementary formulae of $\boldsymbol{Lang}(\Sigma^*)$ need not be computable!)

6.3 Conclusion

In this section, using concepts of realizability, we were able to prove the Main Lemma, and hence the Selection Theorem, without having to assume computability of equality, but at the expense of having to work with an intuitionistic proof system.

Hence the result in this section cannot really be considered a generalization of the Parsons-Mints-Takeuti Theorem.

7 Selection theorem for algebras without computable equality and with classical proof system

7.1 Our aim; Counterexample?

We want to prove the Main Lemma, and hence the Selection Theorem, for algebras without either of the restrictions of the last two sections, *i.e.*, without the computable equality assumption, and without having to work in intuitionistic systems.

We should first ask, however: is the Selection Theorem true without these two restrictions? Here is a proposed counterexample. Consider the algebra $\mathcal{R}$ of reals (§5.3) and the quantifier-free formula

$$P(\mathtt{x},\mathtt{y}) \quad \equiv_{df} \quad (\mathtt{x} \neq 0 \wedge \mathtt{y} = 0) \vee (\mathtt{x} = 0 \wedge \mathtt{y} = 1)$$

where $\mathtt{x},\mathtt{y}\colon \mathsf{real}$. Then

$$\forall \mathtt{x}\, \exists \mathtt{y}\, P(\mathtt{x},\mathtt{y})$$

is classically true and easily provable classically. But the (unique) selection function for this is *not continuous* on $\mathbb{R}$, and hence not PR^* computable on $\mathcal{R}$.

Note, however, that P has a *negated equality*, and is therefore *not* elementary, according to our definition (§4.1(*b*)), or even $\mathbf{\Sigma_1^*}$!

7.2 Solution: extend realizability to sequents

The solution is to *extend* the concept of realizability used in Section 6 to realizability of *sequents*, following [Str03]. So given a sequent

$$\Delta \;\equiv\; P_1, \ldots, P_n$$

of product type $u = u_1 \times \cdots \times u_n$, and a Σ^*-term tuple

$$\bar{r} \;=\; \langle r_0, r_1, \ldots, r_n \rangle$$

of "matching" type $\mathsf{nat} \times u_1, \ldots, u_n$, we define

$$\bar{r} \;\rhd\!\rhd\; \Delta \qquad (\text{“}\bar{r} \text{ realises } \Delta\text{”})$$

to mean

$$(r_0 = 1 \wedge r_1 \rhd P_1) \;\vee\; (r_0 = 2 \wedge r_2 \rhd P_2) \;\vee\; \ldots \;\vee\; (r_0 = n \wedge r_n \rhd P_n)$$

(where '$\rhd$' is defined as in §6.1). Notice that $\bar{r}$ has an initial term r_0 of type nat, followed by a term tuple of the same product type as Δ.

Intuitively, $\bar{r}$ realises Δ (understood disjunctively) by selecting one of the P_i according to the value i of r_0, and then realising it with r_i. We call the term r_0 the *index* of the realiser $\bar{r}$, since it indicates which formula in Δ is actually being realised.

We can now state the current version of the Main Lemma (*cf.* §6.1).

Main Lemma. Suppose the $\mathbf{\Sigma_1^*}$ sequent

$$Q_1, \ldots, Q_m \quad \longmapsto \quad P_1, \ldots, P_n \tag{7.1}$$

is provable in $\mathbf{\Sigma_1^*}$-$\mathsf{Ind}(\Sigma, T)$. Let $Q_1, \ldots, Q_m, P_1, \ldots, P_n$ have types $v_1, \ldots, v_m, w_1, \ldots$ $\ldots, w_n$ respectively, and $\boldsymbol{var}(Q_1, \ldots, Q_m, P_1, \ldots, P_n) \subseteq \mathtt{x} : u$. Let $\mathtt{z}_1, \ldots, \mathtt{z}_m$ be tuples of variables, pairwise disjoint and disjoint from $\mathtt{x}$, with $\mathtt{z}_i \colon v_i$ for $i = 1, \ldots, m$. Then for some tuple of PR* schemes $\alpha \colon u \times v_1 \times \cdots \times v_m \ \to \ \mathsf{nat} \times w_1 \times \cdots \times w_n$,

$$\mathtt{z}_1 \rhd Q_1, \ \ldots, \ \mathtt{z}_m \rhd Q_m \quad \longmapsto \quad \mathsf{f}_\alpha(\mathtt{x}, \mathtt{z}_1, \ldots, \mathtt{z}_m) \rhd\!\rhd (P_1, \ldots, P_n) \tag{7.2}$$

is provable in $\mathbf{\Sigma_1^*}$-$\mathsf{Ind}_\mathsf{i}(\Sigma^*_\alpha, \mathrm{T}{+}E_\alpha)$, where Σ^*_α is the extension of Σ^* with symbols for the function tuple together with their auxiliary functions, and E_α is the equational specification for these functions.

7.3 Proof of Main Lemma

We introduce the following terminology and notation.

(*i*) The sequent (7.1) is said to be *covered by* $\mathtt{x}$ if $\boldsymbol{var}(Q_1, \ldots, Q_m, P_1, \ldots, P_n) \subseteq \mathtt{x}$.

(*ii*) We express (7.2) by saying that f_α *realises the sequent* (7.1) (w.r.t. $\mathtt{x}$).

(*iii*) Suppose $\Gamma \equiv Q_1, \ldots, Q_m$, with $Q_i : v_i$. Then we write $\Gamma : v_1 \times \cdots \times v_m$. If, further, $\mathtt{z} \equiv \mathtt{z}_1, \ldots, \mathtt{z}_m$ with $\mathtt{z}_i : v_i$, then we write '$\mathtt{z} \rhd \Gamma$' for $\mathtt{z}_1 \rhd Q_1$, $\ldots$, $\mathtt{z}_m \rhd Q_m$. Note: this is just a notational shorthand (Γ is read "conjunctively"); it is *not* the same as the new concept '$\mathtt{z} \rhd\!\rhd \Delta$' defined in §7.2 (where Δ is read "disjunctively").

By the Cut Reduction Lemma (in §5.1) we may assume we have a $\mathbf{\Sigma_1^*}$ derivation of (7.1). The required PR* schemes are then constructed by induction on the length of such a derivation.

The *base case* involves initial sequents. By the Remark on Initial Sequents (in §4.2) the initial sequents contain only BU equations, except for axiom (4.5*a*). Hence the result holds trivially for all initial sequents other than (4.5*a*), since by Remark 3 in §6.1, any BU equational sequent can be trivially realised, by (for example) a function tuple of the correct type with default constant value. (Here the Instantiation Assumption on Σ (§2.1) is being used.)

As for the initial sequent (4.5*a*), or rather a substitution instance

$$\longmapsto \quad (t = \mathsf{true}) \ \vee \ (t = \mathsf{false})$$

for any boolean term t with $\boldsymbol{var}(t) \subseteq \mathtt{x} \colon u$ (say), this can be realised by a scheme tuple $\alpha \colon u \ \to \ \mathsf{nat} \times \mathsf{bool}^3$, where

$$\mathsf{f}_\alpha(\mathtt{x}) \ = \ \langle 1, \, (t, \mathsf{true}, \mathsf{true}) \rangle.$$

For the *induction step*, there are different cases according to the last inference of the derivation.

Consider now the three inferences which (explicitly or implicitly) use contraction in the succedent: Contr:R, $\wedge$R and $\vee$L. First, Contr:R:

$$\frac{\Gamma \;\longmapsto\; P, P, \Delta}{\Gamma \;\longmapsto\; P, \;\Delta} \tag{7.3}$$

Suppose the conclusion is covered by $\mathtt{x} : u$. Then the premiss is also covered by $\mathtt{x}$. Assume $\Gamma : v,\ P : w_0$ and $\Delta : w$.

By induction hypothesis, there is a PR* scheme tuple $\alpha\colon u \times v \;\to\; \mathsf{nat} \times w_0^2 \times w$ which realises the premiss of (7.3), *i.e.*, such that

$$\mathtt{z} \;\rhd\; \Gamma \;\longmapsto\; \mathsf{f}_\alpha(\mathtt{x}, \mathtt{z}) \;\rhd\!\rhd\; P, P, \Delta$$

is provable. Put $\mathsf{f}_\alpha(\mathtt{x}, \mathtt{z}) = \langle r_0, r_1, r_2, \bar{r}\rangle$ where the realising terms $r_0 : \mathsf{nat},\ r_1 : v,\ r_2 : v$ and $\bar{r} : w$ represent PR* functions applied to $\mathtt{x}$,$\mathtt{z}$.

We can then easily construct a PR* scheme tuple $\beta\colon u \times v \;\to\; \mathsf{nat} \times w_0 \times w$ with

$$\mathsf{f}_\beta(\mathtt{x}, \mathtt{z}) \;=\; \langle r_0', r_1', \bar{r}\rangle$$

where

$$r_0' \;=\; \begin{cases} 1 & \text{if } \; r_0 = 1 \vee r_0 = 2 \\ r_0 - 1 & \text{if } \; r_0 > 2 \end{cases}$$

and

$$r_1' \;=\; \begin{cases} r_1 & \text{if } \; r_0 = 1 \\ r_2 & \text{if } \; r_0 = 2 \\ \text{arbitrary} & \text{if } \; r_0 > 2. \end{cases}$$

Then f_β realises the conclusion of (7.3).

Remark: The contracted formula P is realised in the conclusion by either r_1 or r_2 (which realised the two occurrences of P in the premisses) depending on the value of the index r_0, and hence of r_0'.

Suppose now the last inference is $\wedge$R:

$$\frac{\Gamma \;\longmapsto\; P_1, \Delta \qquad \Gamma \;\longmapsto\; P_2, \Delta}{\Gamma \;\longmapsto\; P_1 \wedge P_2, \Delta}. \tag{7.4}$$

Suppose the conclusion is covered by $\mathtt{x} : u$. Then the premisses are also covered by $\mathtt{x}$. Assume $\Gamma : v,\ P_1 : w_1,\ P_2 : w_2$ and $\Delta : w$.

By induction hypothesis there are PR* scheme tuples $\alpha_1 : u \times v \;\to\; \mathsf{nat} \times w_1 \times w$ and $\alpha_2 : u \times v \;\to\; \mathsf{nat} \times w_2 \times w$ which realise the premisses of (7.4), *i.e.*, such that

$$\begin{aligned} \mathtt{z} \rhd \Gamma \;&\longmapsto\; \mathsf{f}_{\alpha_1}(\mathtt{x}, \mathtt{z}) \rhd\!\rhd P_1, \Delta \\ \mathtt{z} \rhd \Gamma \;&\longmapsto\; \mathsf{f}_{\alpha_2}(\mathtt{x}, \mathtt{z}) \rhd\!\rhd P_2, \Delta \end{aligned}$$

are provable. Put

$$\begin{aligned} \mathsf{f}_{\alpha_1}(\mathtt{x},\mathtt{z}) &= \langle r_0^1,\ r_1^1,\ \bar{r}^1\rangle \\ \mathsf{f}_{\alpha_2}(\mathtt{x},\mathtt{z}) &= \langle r_0^2,\ r_1^2,\ \bar{r}^2\rangle \end{aligned}$$

where $r_0^i : \mathsf{nat},\ r_1^i : w_i$ and $\bar{r}^i : w\ \ (i = 1, 2)$. We can then construct a PR* scheme tuple $\beta\colon u \times v \ \to\ \mathsf{nat} \times (w_1 \times w_2) \times w$ where

$$\mathsf{f}_\beta(\mathtt{x},\mathtt{z}) \ = \ \langle r_0,\ r_1,\ \bar{r}\rangle$$

with

$$r_0 \ = \ \begin{cases} 1 & \text{if } \ r_0^1 = 1 \ \wedge \ r_0^2 = 1 \\ r_0^1 & \text{if } \ r_0^1 > 1 \\ r_0^2 & \text{if } \ r_0^1 = 1 \ \wedge \ r_0^2 > 1 \end{cases}$$

and

$$r_1 \ = \ (r_1^1,\ r_1^2)$$

and

$$\bar{r} \ = \ \begin{cases} \bar{r}^{\,1} & \text{if } \ r_0^1 > 1 \\ \bar{r}^{\,2} & \text{if } \ r_0^1 = 1 \ \wedge \ r_0^2 > 1 \\ \text{arbitrary} & \text{if } \ r_0^1 = 1 \ \wedge \ r_0^2 = 1. \end{cases}$$

Then f_β realises the conclusion of (7.4).

Remark: The side formulas in the succedent, *i.e.*, the formulas in Δ, are implicitly contracted. Each one is realised by the corresponding term in either $\bar{r}^{\,1}$ or $\bar{r}^{\,2}$, depending on the values of the indices r_0^1 and r_0^2. Note that in the absence of such side formulas, *i.e.*, if Δ is empty (as in the intuitionistic system), the construction of the scheme β from α is very simple.

The remaining inference that uses contraction in the succedent is $\vee\mathsf{L}$:

$$\frac{\Gamma,\, Q_1 \ \longmapsto\ \Delta \qquad \Gamma,\, Q_2 \ \longmapsto\ \Delta}{\Gamma,\, Q_1 \vee Q_2 \ \longmapsto\ \Delta}\,.$$

Here the construction of a realiser for the conclusion from realisers for the premisses is almost exactly the same as in the intuitionistic case (§6.2). The only difference is that the string '$\rhd P$', which occur in 3 places (at the right end of the sequents (6.2) and (6.3)), is replaced by '$\rhd\rhd\Delta$'.

In the cases *thinning*, *interchange* and Contr:L, a realiser for the conclusion can be obtained easily from a realiser for the premiss.

Consider now the logical inferences. Since the derivation is $\mathbf{\Sigma_1^*}$, there are no '$\to$' or '$\forall$' inferences.

We have dealt with $\wedge$R and $\vee$L above. The cases $\wedge$L is quite simple. Consider now $\vee$R:

$$\frac{\Gamma \ \longmapsto\ P_1, P_2,\ \Delta}{\Gamma \ \longmapsto\ P_1 \vee P_2, \Delta} \tag{7.5}$$

Suppose the conclusion is covered by $\mathtt{x}$. Then so is the premiss.

By induction hypothesis there is a scheme tuple α which realises the premiss of (7.5), *i.e.*, such that

$$\mathtt{z} \rhd \Gamma \;\longmapsto\; \mathsf{f}_\alpha(\mathtt{x},\mathtt{z}) \rhd\!\rhd P_1, P_2, \Delta.$$

Put

$$\mathsf{f}_\alpha(\mathtt{z}) \;=\; \langle r_0,\, r_1,\, r_2,\, \bar{r}\rangle.$$

Then we can construct a scheme tuple β such that

$$\mathsf{f}_\beta(\mathtt{x},\mathtt{z}) \;=\; \langle r_0',\; \langle r_{\mathsf{B}},\, r_1,\, r_2\rangle,\; \bar{r}\rangle$$

where

$$r_0' \;=\; \begin{cases} 1 & \text{if } \; r_0 = 1 \vee r_0 = 2 \\ r_0 - 1 & \text{if } \; r_0 > 2 \end{cases}$$

and $r_{\mathsf{B}} : \mathsf{bool}$ with

$$r_{\mathsf{B}} \;=\; \begin{cases} \mathsf{true} & \text{if } \; r_0 = 1 \\ \mathsf{false} & \text{if } \; r_0 = 2 \\ \text{arbitrary} & \text{if } \; r_0 > 2. \end{cases}$$

Then f_β realises the conclusion of (7.5).

Suppose the last inference is $\forall_{\mathsf{b}} R$:

$$\frac{\Gamma,\, \mathtt{a} < t \;\longmapsto\; P(\mathtt{a}),\, \Delta}{\Gamma \;\longmapsto\; \forall \mathtt{k} < t\, P(\mathtt{k}),\, \Delta} \tag{7.6}$$

where the eigenvariable $\mathtt{a} : \mathsf{nat}$ does not occur in the conclusion. Suppose the conclusion is covered by $\mathtt{x} : u$. Then the premiss is covered by $(\mathtt{x},\mathtt{a}) : u \times \mathsf{nat}$. Assume $\Gamma : v$, $P : w_0$ and $\Delta : w$.

By induction hypothesis there is a PR* scheme tuple $\alpha : u \times \mathsf{nat} \times v \times \mathsf{bool} \;\to\; \mathsf{nat} \times w_0 \times w$ which realises the premiss of (7.6), *i.e.*,

$$\mathtt{z} \rhd \Gamma,\;\; \mathtt{z}_0 \rhd \mathtt{a} < t \;\longmapsto\; \mathsf{f}_\alpha(\mathtt{x},\mathtt{a},\mathtt{z},\mathtt{z}_0) \rhd\!\rhd P(\mathtt{a}), \Delta.$$

Note that $\mathtt{a} < t$ means $\mathsf{less}_{\mathsf{nat}}(\mathtt{a}, t) = \mathsf{true}$, which is trivially realised by anything of type bool. Put

$$\mathsf{f}_\alpha(\mathtt{a},\mathtt{x},\mathtt{z},\mathtt{z}_0) \;=\; \langle r_0(\mathtt{a}),\, r_1(\mathtt{a}),\, \bar{r}(\mathtt{a})\rangle$$

(making explicit the dependence of the realising terms on the eigenvariable $\mathtt{a}$). We can then construct a scheme tuple $\beta : u \times v \;\to\; \mathsf{nat} \times w_0^* \times w$ (note the array type in the range!) such that

$$\mathsf{f}_\beta(\mathtt{x},\mathtt{z}) \;=\; \langle r_0',\, r_1^*,\, \bar{r}\,'\rangle$$

where $r_0' : \mathsf{nat}$, $r_1^* : w_0^*$ and $\bar{r}\,' : w$ are defined as follows:

Case 1: For all $k < t$, $r_0(k) = 1$. Then define

$$r_0' \;=\; 1$$

and $\bar{r}\,'$ arbitrarily (for example, $\bar{r}\,' = \bar{r}(0)$).

Case 2: For some $k_0 < t,\ r_0(k_0) > 1$. Then define

$$\begin{aligned} r_0' &= r_0(k_0) \\ \bar{r}' &= \bar{r}(k_0). \end{aligned}$$

And in both cases, define $r_1^*\colon w_0^*$ by

$$r_1^*[k] = r_1(k) \qquad \text{for all } k < t.$$

Then f_β realises the conclusion of (7.6).

Remarks: (1) The two cases above are PR* distinguishable, being based on bounded quantification, which is primitive recursively decidable. (2) The choice of k_0 in Case 2 is not important, since the formulae in Δ do not contain the eigenvariable $\mathtt{a}$, and hence their realizability by $\bar{r}(\mathtt{a})$ does not depend on the value k of $\mathtt{a}$.

Now suppose the last inference is $\forall_{\mathsf{b}} L$:

$$\frac{\Gamma \ \longmapsto\ t_0 < t, \Delta \qquad \Gamma, Q(t_0) \ \longmapsto\ \Delta}{\Gamma, \forall \mathtt{k} < t\, Q(\mathtt{k}) \ \longmapsto\ \Delta}$$

Let $\hat{t}_0$ be the term formed from t_0 by replacing all variables in t_0 which are *not* free in the conclusion by default terms of the same sort. (Here the Instantiation Assumption is being used.) Then the derivation can be easily modified so as to end in the inference

$$\frac{\Gamma \ \longmapsto\ \hat{t}_0 < t, \Delta \qquad \Gamma, Q(\hat{t}_0) \ \longmapsto\ \Delta}{\Gamma, \forall \mathtt{k} < t\, Q(\mathtt{k}) \ \longmapsto\ \Delta} \tag{7.7}$$

with the same conclusion, but with the additional property that if $\mathtt{x}\colon u$ covers the conclusion, then it also covers both premisses. Assume $\Gamma : v,\ Q : v_0$ and $\Delta : w$.

By induction hypothesis, there are scheme tuples $\alpha_1\colon u \times v \ \to\ \mathsf{nat} \times \mathsf{bool} \times w$ and $\alpha_2\colon u \times v \times v_0 \ \to\ \mathsf{nat} \times w$ such that f_{α_1} and f_{α_2} realise the two premisses of (7.7), *i.e.*:

$$\begin{aligned} \mathtt{z} \rhd \Gamma \ &\longmapsto\ \mathsf{f}_{\alpha_1}(\mathtt{x}, \mathtt{z}) \rhd\!\rhd \hat{t}_0 < t, \Delta \\ \mathtt{z} \rhd \Gamma,\ \mathtt{z}_0 \rhd Q(\hat{t}_0) \ &\longmapsto\ \mathsf{f}_{\alpha_2}(\mathtt{x}, \mathtt{z}, \mathtt{z}_0) \rhd\!\rhd \Delta. \end{aligned}$$

Put

$$\begin{aligned} \mathsf{f}_{\alpha_1}(\mathtt{x}, \mathtt{z}) &= \langle r_0^1,\, \mathsf{true},\, \bar{r}^{\,1} \rangle \\ \mathsf{f}_{\alpha_2}(\mathtt{x}, \mathtt{z}, \mathtt{z}_0) &= \langle r_0^2(\mathtt{z}_0),\, \bar{r}^{\,2}(\mathtt{z}_0) \rangle. \end{aligned}$$

(making explicit the dependence of the realising terms on the realiser $\mathtt{z}_0$ of $Q(\hat{t}_0)$). Note again that the atomic formula $\hat{t}_0 < t$ is trivially realised by anything of type bool.) Now we can construct a scheme tuple $\beta\colon u \times v \times v_0^* \ \to\ \mathsf{nat} \times w$ (note the array type in the domain!) with

$$\mathsf{f}_\beta(\mathtt{x}, \mathtt{z}, \mathtt{z}_0^*) = \langle r_0, \bar{r} \rangle$$

where

$$r_0 \;=\; \begin{cases} r_0^2(\mathsf{z}_0^*[\hat{t}_0]) & \text{if } \; r_0^1 = 1 \\ r_0^1 & \text{if } \; r_0^1 > 1 \end{cases}$$

and

$$\bar{r} \;=\; \begin{cases} \bar{r}^{\,2}(\mathsf{z}_0^*[\hat{t}_0]) & \text{if } \; r_0^1 = 1 \\ \bar{r}^{\,1} & \text{if } \; r_0^1 > 1 \end{cases}$$

Then f_β realises the conclusion of (7.7).

Suppose next that the last inference is $\exists R$:

$$\frac{\Gamma \;\longmapsto\; P(t),\ \Delta}{\Gamma \;\longmapsto\; \exists \mathsf{y} P(\mathsf{y}),\ \Delta}.$$

As with $\forall_{\mathsf{b}} L$, let $\hat{t}$ be the term formed from t by replacing all variables in t which are *not* free in the conclusion by default terms of the same sort. (Here again the Instantiation Assumption is being used.) Then the derivation can be easily modified so as to end in the inference

$$\frac{\Gamma \;\longmapsto\; P(\hat{t}),\ \Delta}{\Gamma \;\longmapsto\; \exists \mathsf{y} P(\mathsf{y}),\ \Delta}. \tag{7.8}$$

with the same conclusion, but with the additional property that if $\mathtt{x}\colon u$ covers the conclusion, then it also covers the premiss.

By induction hypothesis, there is a PR* scheme tuple α such that f_α realises the premiss of (7.8), *i.e.*,

$$\mathsf{z} \rhd \Gamma \;\longmapsto\; \mathsf{f}_\alpha(\mathtt{x}, \mathsf{z}) \rhd\!\rhd P(\hat{t}), \Delta.$$

Put

$$\mathsf{f}_\alpha(\mathtt{x}, \mathsf{z}) \;=\; \langle r_0,\, r_1,\, \bar{r} \rangle.$$

We can then construct a scheme tuple β such that

$$\mathsf{f}_\beta(\mathtt{x}, \mathsf{z}) \;=\; \langle r_0,\ \langle \hat{t},\, r_1 \rangle,\ \bar{r} \rangle.$$

Then f_β realises the conclusion of (7.8).

Suppose next that the last inference is $\exists L$:

$$\frac{\Gamma,\quad Q(\mathtt{a}) \;\longmapsto\; \Delta}{\Gamma,\ \exists \mathsf{y}\, Q(\mathsf{y}) \;\longmapsto\; \Delta} \tag{7.9}$$

where the eigenvariable $\mathtt{a}$ does not occur in the conclusion. Assume $\Gamma\colon v$, $Q\colon v_0$, $\Delta\colon w$ and $\mathtt{y}\colon s$. Assume also that the conclusion is covered by $\mathtt{x}\colon u$. Then the premiss is covered by $(\mathtt{x}, \mathtt{a})\colon u \times s$.

By induction hypothesis, there is a scheme tuple $\alpha\colon (u \times s) \times v \times v_0 \;\to\; \mathsf{nat} \times w$ which realises the premiss of (7.9), *i.e.*,

$$\mathsf{z} \rhd \Gamma,\ \mathsf{z}_0 \rhd Q(\mathtt{a}) \;\longmapsto\; \mathsf{f}_\alpha((\mathtt{x}, \mathtt{a}), \mathsf{z}, \mathsf{z}_0) \rhd\!\rhd \Delta.$$

So define $\beta\colon u \times v \times (s \times v_0) \;\to\; \mathsf{nat} \times w$ by

$$\mathsf{f}_\beta(\mathtt{x}, \mathtt{z}, (\mathtt{a}, \mathtt{z}_0)) \;=\; \mathsf{f}_\alpha((\mathtt{x}, \mathtt{a}), \mathtt{z}, \mathtt{z}_0).$$

Then f_β realises the conclusion of (7.9), *i.e.*,

$$\mathtt{z} \rhd \Gamma,\ (\mathtt{a}, \mathtt{z}_0) \rhd \exists \mathtt{y}\, Q(\mathtt{y}) \quad\longmapsto\quad \mathsf{f}_\beta((\mathtt{x}, \mathtt{z}, (\mathtt{a}, \mathtt{z}_0)) \rhd\!\rhd \Delta.$$

Comment: Notice that f_β is essentially the same as f_α, except that the eigenvariable $\mathtt{a}$, which is one of the free variables of the sequent in the premiss, is re-interpreted as part of the realiser of $\exists \mathtt{y}\, Q(\mathtt{y})$ in the conclusion.

Now suppose the last inference is a *cut*, which we take, for convenience, in the following form:

$$\frac{\Gamma \;\longmapsto\; P, \Delta \qquad \Gamma, P \;\longmapsto\; \Delta}{\Gamma \;\longmapsto\; \Delta}.$$

Since the derivation is $\mathbf{\Sigma_1^*}$ by the Cut Reduction Lemma, the cut formula P is $\mathbf{\Sigma_1^*}$. Now (as with $\forall_{\mathsf{b}} L$ and $\exists R$) let $\widehat{P}$ be the formula formed from P by replacing all variables in P which are *not* free in the conclusion by default terms of the same sort. (Here again the Instantiation Assumption is being used.) Then the derivation can be simply modified so as to end in the cut

$$\frac{\Gamma \;\longmapsto\; \widehat{P}, \Delta \qquad \Gamma, \widehat{P} \;\longmapsto\; \Delta}{\Gamma \;\longmapsto\; \Delta}. \tag{7.10}$$

with the same conclusion, but with the additional property that if $\mathtt{x}\colon u$ covers the conclusion, then it also covers the premisses.

Assume that $\Gamma\colon v$, $\Delta\colon w$ and $\widehat{P}\colon v_0$. By induction hypothesis, there are schemes $\alpha\colon u \times v \;\to\; \mathsf{nat} \times v_0 \times w$ and $\beta\colon u \times v \times v_0 \;\to\; \mathsf{nat} \times w$ which realise the two premisses, *i.e.*,

$$\begin{aligned} \mathtt{z} \rhd \Gamma \quad&\longmapsto\quad \mathsf{f}_\alpha(\mathtt{x}, \mathtt{z}) \rhd\!\rhd \widehat{P}, \Delta \\ \mathtt{z} \rhd \Gamma,\ \mathtt{z}_0 \rhd \widehat{P} \quad&\longmapsto\quad \mathsf{f}_\beta(\mathtt{x}, \mathtt{z}, \mathtt{z}_0) \rhd\!\rhd \Delta. \end{aligned}$$

We can construct a scheme tuple $\gamma\colon u \times v \;\to\; \mathsf{nat} \times w$ as follows. Put

$$\mathsf{f}_\alpha(\mathtt{x}, \mathtt{z}) \;=\; \langle r_0,\, r_1,\, \bar{r} \rangle.$$

There are two cases.

Case 1: $r_0 > 1$. Then we let

$$\mathsf{f}_\gamma(\mathtt{x}, \mathtt{z}) \;=\; \langle r_0,\, \bar{r} \rangle.$$

Case 2: $r_0 = 1$. Then we let

$$\mathsf{f}_\gamma(\mathtt{x}, \mathtt{z}) \;=\; \mathsf{f}_\beta(\mathtt{x},\, \mathtt{z},\, r_1).$$

Then f_γ realises the conclusion of (7.10).

Comment: So the conclusion of a *cut* is realised essentially by *composition* of the realisers of the premisses.

Suppose, finally, that the last inference is $\mathbf{\Sigma_1^*}$ *induction* (repeating (4.4) here for convenience):

$$\frac{\Gamma,\ P(\mathsf{a}) \ \longmapsto\ P(\mathsf{Sa}),\ \Delta}{\Gamma,\ P(0) \ \longmapsto\ P(t),\ \Delta}. \tag{7.11}$$

where the induction formula $P(\mathsf{a})$ is $\mathbf{\Sigma_1^*}$, and the induction variable $\mathtt{a}\colon \mathsf{nat}$ does not occur in Γ, Δ or $P(0)$. Suppose the conclusion is covered by $\mathtt{x}\colon u$. Then the premiss is covered by $(\mathtt{x},\mathtt{a})\colon u \times \mathsf{nat}$. Assume $\Gamma\colon v$, $P(\mathsf{a})\colon v_0$ and $\Delta\colon w$.

By induction hypothesis there is a scheme $\alpha\colon \mathsf{nat} \times u \ \times v \times v_0 \ \to\ \mathsf{nat} \times v_0 \times w$ which realises the premiss, *i.e.*,

$$\mathtt{z} \rhd \Gamma,\ \mathtt{z}_0 \rhd P(\mathsf{a}) \ \longmapsto\ \mathsf{f}_\alpha(\mathtt{x},\mathtt{a},\mathtt{z},\mathtt{z}_0) \rhd\!\rhd P(\mathsf{Sa}),\Delta. \tag{7.12}$$

Put

$$\mathsf{f}_\alpha(\mathtt{x},\mathtt{a},\mathtt{z},\mathtt{z}_0) \ =\ \langle r_0(\mathtt{a},\mathtt{z}_0),\ r_1(\mathtt{a},\mathtt{z}_0),\ r_2(\mathtt{a},\mathtt{z}_0),\ \ldots\rangle, \tag{7.13}$$

(making explicit the dependence of the realising terms $r_0, r_1, r_2, \ldots$ on the variables $\mathtt{a}$ and $\mathtt{z}_0$). Now we construct a scheme $\beta\colon u \ \times v \times v_0 \ \to\ \mathsf{nat} \times v_0 \times w$ such that

$$\mathsf{f}_\beta(\mathtt{x},\mathtt{z},\mathtt{z}_0) \ =\ \langle r'_0(t,\mathtt{z}_0),\ r'_1(t,\mathtt{z}_0),\ r'_2(t,\mathtt{z}_0),\ \ldots\rangle$$

where the realisers $r'_0, r'_1, r'_2, \ldots$ are defined by *simultaneous primitive recursion*:

Base case:

$$\begin{aligned} r'_i(0,\mathtt{z}_0) \ &=\ r_i(0,\mathtt{z}_0) \qquad \text{for } i \neq 1 \\ r'_1(0,\mathtt{z}_0) \ &=\ \mathtt{z}_0. \end{aligned}$$

Recursion step: For all $i = 0, 1, 2, \ldots$:

$$r'_i(\mathtt{n}+1,\mathtt{z}_0) \ =\ \begin{cases} r'_i(\mathtt{n},\mathtt{z}_0) & \text{if } \ r'_0(\mathtt{n},\mathtt{z}_0) > 1 \\ r_i(\mathtt{n},\ r'_1(\mathtt{n},\mathtt{z}_0)) & \text{if } \ r'_0(\mathtt{n},\mathtt{z}_0) = 1 \quad \text{(the "interesting case").} \end{cases}$$

Thus, as soon as the index points to a realiser in Δ, *i.e.*, $r'_0(\mathtt{n},\mathtt{z}_0) > 1$, everything remains constant; otherwise we carry on inductively as expected.

Then f_β realises the conclusion of (7.11), *i.e.*,

$$\mathtt{z} \rhd \Gamma,\ \mathtt{z}_0 \rhd P(0) \ \longmapsto\ \mathsf{f}_\beta(\mathtt{x},\mathtt{z},\mathtt{z}_0) \rhd\!\rhd P(t),\Delta$$

is provable by $\mathbf{\Sigma_1^*}$ -induction on (the value of) t.

This concludes the proof of the Main Lemma, and hence of the Selection Theorem.

8 Some concluding remarks

8.1 The use of starred sorts and systems

Note first that the use of starred (or array) sorts is strongly connected with the use of the *bounded universal quantifier*. For (in one direction)

(1) BU conditional equations are used in the axiomatisation of array equality (see equation (4.6));

and (in the other direction)

(2) an existentially quantified variable changes to a starred sort when permuting with a BU quantifier in the transformation to prenex form (§4.1, Lemma 1);

(3) an array term is needed for the realisation of BU quantification (§6.1, Def. 2(iv)); and hence

(4) an array-valued function is needed for the conclusion of a $\forall_{\mathsf{b}} R$ inference, either as a *selector* for the prenex form (in Section 5) or as a *realiser* (in Section 7; see (7.6)).

In fact, the use of starred sorts and systems in this paper (for example, $\mathbf{\Sigma_1^*}$-$\mathsf{Ind}(\Sigma, T)$ instead of $\mathbf{\Sigma_1}$-$\mathsf{Ind}(\Sigma, T)$), and the use of BU quantification (as a clause in the definition of $\mathbf{\Sigma_1^*}$ formulae, and as a primitive inference rule) could *both have been omitted*. The main results of this paper, *i.e.*, the Selection and Provable Selection Theorems, as used in Sections 5, 6 and 7, could all have been formulated in a "starless" and "BU-less" form.

However, working with array sorts, and BU quantification, allowed the results to be presented in a more general setting.

For example, the construction of the $\mathbf{\Sigma_1^*}$ computation predicate P_α, discussed in §4.6, needs starred sorts, even for PR schemes α. Hence the Totality Lemma in §4.6 (even for PR schemes), and the Corollary in §5.1 (even for μPR schemes) both need starred sorts, even for their formulation.

8.2 Total vs partial algebras

In this paper we have considered only total algebras. This is a real restriction, since partial basic functions occur quite naturally in topological algebras; consider, for example, the algebra $\mathcal{R}$ of reals (1.3) augmented with continuous partial operators of division, equality and order [TZ04]. To extend the current theory to such partial algebras would entail extending the proof theory used here to a logic of partial terms or definedness (see, *e.g.*, [Bee85, pp. 97–99] and [Fef95]). This is likely to be a major undertaking, but one worth pursuing.

8.3 Other functional interpretations

It would be interesting to know whether the results of this paper could also be obtained using other types of functional interpretation; for example, a version of Gödel's Dialectica interpretation [AF98], or a Herbrand-style interpretation [Bus98b, §2.5], versions of which have also been applied to fragments of arithmetic [Sie91, Koh92], or the (related) "witness function" methods of Buss [Bus86, Bus94, Bus98a].

Recall that Herbrand interpretations are typically applied to classical systems, to transform proofs of existential statements

$$\exists \mathtt{y} P(\mathtt{x}, \mathtt{y}) \tag{8.1}$$

with P quantifier-free, to proofs of finite disjunctions

$$P(\mathtt{x}, t_1) \vee \ldots \vee P(\mathtt{x}, t_n) \tag{8.2}$$

($n \geq 1$, t_i term tuples containing $\mathtt{x}$). Now to construct a selection function for (8.1), assuming the system has decidable atomic formulae, (8.2) can be contracted to a statement

$$P(\mathtt{x}, t) \tag{8.3}$$

for a single term tuple t constructed with a simple definition by cases, as in Section 5. However, without decidability of atomic formulae, it is not at all apparent how to proceed from (8.2) to (8.3) (or, for that matter, to interpret $\mathbf{\Sigma_1}$ induction suitably) without analysing the proof of (8.2) (or (8.1)) by a *realizability* interpretation, as in Section 7.

Acknowledgments

The authors wish to thank two anonymous referees for helpful suggestions with an earlier version of this paper. The research of the second author was supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

References

[AF98] J. Avigad and S. Feferman. Gödel's functional ("Dialectica") interpretation. In S.R. Buss, editor, *Handbook of Proof Theory*, pages 337–405. Elsevier, 1998.

[Bee85] M. Beeson. *Foundations of Constructive Mathematics*. Springer-Verlag, 1985.

[Bus86] S. R. Buss. *Bounded Arithmetic*. Bibliopolis, Napoli, 1986.

[Bus94] S.R. Buss. The witness function method and fragments of Peano arithmetic. In D. Prawitz, B. Skyrms, and D. Westerståhl, editors, *Proceedings of the Ninth International Congress on Logic, Methodology and Philosophy of Science, Uppsala, August 1991*, pages 29–68. Elsevier, 1994.

[Bus98a] S.R. Buss. First-order proof theory of arithmetic. In S.R. Buss, editor, *Handbook of Proof Theory*, pages 79–147. Elsevier, 1998.

[Bus98b] S.R. Buss. An introduction to proof theory. In S.R. Buss, editor, *Handbook of Proof Theory*, pages 1–78. Elsevier, 1998.

[Can02] A. Cantini. Polytime, combinatory logic and safe induction. *Archive for Mathematical Logic*, 41(2):169–189, 2002.

[Can05] A. Cantini. Choice and uniformity in weak applicative theories. In M. Baaz, Sy Friedman, and J. Krajíček, editors, *Logic Colloquium '01*, volume 20 of *Lecture Notes in Logic*, pages 108–138. Association for Symbolic Logic, 2005.

[Fef75] S. Feferman. A language and axioms for explicit mathematics. In J.N. Crossley, editor, *Algebra and Logic*, volume 450 of *Lecture Notes in Mathematics*, pages 87–139. Springer, Berlin, 1975.

[Fef79] S. Feferman. Constructive theories of functions and classes. In M. Boffa, D. van Dalen, and K. McAloon, editors, *Logic Colloquium '78*, pages 159–224. North Holland, Amsterdam, 1979.

[Fef95] S. Feferman. Definedness. *Erkenntnis*, 43:295–320, 1995.

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

[Kle45] S.C. Kleene. On the interpretation of intuitionistic number theory. *Journal of Symbolic Logic*, 10:109–124, 1945.

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

[Koh92] U. Kohlenbach. Remarks on Herbrand normal forms and Herbrand realizations. *Archive for Mathematical Logic*, 301:305–317, 1992.

[Kre71] G. Kreisel. Some reasons for generalizing recursion theory. In R.O. Gandy and C.M.E. Yates, editors, *Logic Colloquium '69*, pages 139–198. North Holland, 1971.

[Lei94] D. Leivant. A foundational delineation of poly-time. *Information and Computation*, 110:391–420, 1994.

[Min73] G. Mints. Quantifier-free and one-quantifier systems. *Journal of Soviet Mathematics*, 1:71–84, 1973.

[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.

[Par71] C. Parsons. On a number theoretic choice scheme II. *Journal of Symbolic Logic*, 36:587, 1971. (Abstract).

[Par72] C. Parsons. On n-quantifier induction. *Journal of Symbolic Logic*, 37:466–482, 1972.

[Sch95] A. Schlüter. An extension of Leivant's characterization of polytime by predicative arithmetic. Preprint, Stanford University, 1995.

[Sie91] W. Sieg. Herband analyses. *Archive for Mathematical Logic*, 30:409–441, 1991.

[Str03] T. Strahm. Theories with self-application and computational complexity. *Information and Computation*, 185:263–297, 2003.

[Str04] T. Strahm. A proof-theoretic characterization of the basic feasible functionals. *Theoretical Computer Science*, 329:159–176, 2004.

[Tak75] G. Takeuti. *Proof Theory*. North Holland, 1st edition, 1975. 2nd edition, 1987.

[Tro93] A.S. Troelstra, editor. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis (Second corrected edition)*. Technical Notes X-93-05. Institute for Logic, Language and Computation, Amsterdam, 1993. Originally published as volume 344 of *Lecture Notes in Mathematics*, Springer-Verlag, 1973.

[Tro98] A.S. Troelstra. Realizability. In S.R. Buss, editor, *Handbook of Proof Theory*, pages 407–473. Elsevier, 1998.

[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.

[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.

[TZ02] J.V. Tucker and J.I. Zucker. Abstract computability and algebraic specification. *ACM Transactions on Computational Logic*, 3:279–333, 2002.

[TZ04] J.V. Tucker and J.I. Zucker. Abstract versus concrete computation on metric partial algebras. *ACM Transactions on Computational Logic*, 5:611–668, 2004.

[TZ05] J.V. Tucker and J.I. Zucker. Computable total functions, algebraic specifications and dynamical systems. *Journal of Logic and Algebraic Programming*, 62:71–108, 2005.

[Zuc06] J.I. Zucker. Primitive recursive selection functions over abstract algebras. In A. Beckmann, U. Berger, B. Löwe, and J.V. Tucker, editors, *Logical Approaches to Computational Barriers: Second Conference on Computability in Europe, CiE 2006, Swansea, UK, June/July 2006: Proceedings*, volume 3988 of *Lecture Notes in Computer Science*, pages 595–606. Springer-Verlag, 2006.