Eliminating Skolem Functions in Peano Arithmetic with Interactive Realizability

Federico Aschieri[1], Margherita Zorzi[2*]

[1]**Laboratoire PPS, équipe PI.R2,**
Université Paris 7, INRIA & CNRS

[2] **Laboratoire d'Informatique de Paris-Nord**
UMR CNRS 7030
Institut Galilée – Université Paris-Nord

Abstract

We present a new syntactical proof that first-order Peano Arithmetic with Skolem axioms is conservative over Peano Arithmetic alone for arithmetical formulas. This result – which shows that the Excluded Middle principle can be used to eliminate Skolem functions – has been previously proved by other techniques, among them the epsilon substitution method and forcing. In this paper, we employ Interactive Realizability, a computational semantics for Peano Arithmetic which extends Kreisel's modified realizability to the classical case.

1 Introduction

For a long time it has been known that intuitionistic realizability can be used as a flexible tool for obtaining a wealth of unprovability, conservativity and proof-theoretic results [22, 24]. As title of example, with Kreisel's modified realizability [16], one can show the unprovability of Markov Principle in Heyting Arithmetic in all finite types (HA^ω) and the conservativity of HA^ω with the Axiom of Choice (AC) over HA^ω for negative formulas. In both cases, one starts by showing that any formula provable in one of those systems can be shown to be realizable in HA^ω. In the first case, one proves that the realizability of Markov Principle implies the solvability of the Halting Problem, and concludes that Markov Principle is unprovable in HA^ω. In the second, one exploits the fact that the assertion "t realizes A" is exactly the formula A when A is negative and concludes that HA^ω proves A.

The situation in classical logic has been very different: for a long time it did not exist any realizability notion suitable to interpret directly classical proofs, let alone proving independence or conservation results. However, recently several classical realizability interpretations have been put forward. Among them: Krivine's classical realizability [17], which has been shown in [18] to yield striking unprovability results in Zermelo-Fraenkel set theory, and Interactive realizability [1, 4, 6, 7], which has been shown in [3, 6] to provide conservation results for Π^0_2-formulas.

Being a tool for extracting programs from proofs, it is however quite natural that Interactive realizability is capable of producing Π^0_2-conservativity results. The aim of this paper is to prove that Interactive realizability can as well be used to prove other conservativity results. In particular, let us consider first-order classical Peano Arithmetic PA, which is $\mathsf{HA}+\mathsf{EM}$, where EM is the excluded middle over arithmetical formulas. Then we give a new syntactic proof that PA with the Skolem axiom scheme SK is conservative over PA for arithmetical formulas – a result first syntactically proven by Hilbert and Bernays [15] by means

* Supported by ANR COMPLICE project (Implicit Computational Complexity, Concurrency and Extraction), ref.: ANR-08-BLANC-0211-01.

of the epsilon substitution method. The result is particularly interesting since it implies that classical choice principles can be eliminated by using the excluded middle alone. The structure of our proof resembles the pattern of the intuitionistic-realizability conservation proofs we have sketched above and allows to obtain a stronger result. Namely, we shall show that if an arithmetical formula A is provable in $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$, then the assertion "$t$ realizes A" is provable in HA^ω alone. Afterwards, we shall show the provability in $\mathsf{HA}^\omega + \mathsf{EM}$ of the assertion "(t realizes A) implies A" and thus conclude that $\mathsf{HA}^\omega + \mathsf{EM}$ proves A. Since this latter system is conservative over PA for arithmetical formulas, we obtain the result.

In our opinion, there are at least two reasons our proof technique is interesting. First, it does not lead to an exponential increase in the size of the proof, when passing from a proof in $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$ to a corresponding proof in $\mathsf{HA} + \mathsf{EM}$: our transformation is polynomial. To the best of our knowledge, there is only another method that does equally well, which is Avigad's [9]. The technique of Avigad is related to ours since it uses the method of forcing, in which the conditions are finite approximations of the Skolem functions used in the proof. With forcing one avoids speaking about infinite non-computable objects (i.e. the Skolem functions) and can approximate the original proof. Avigad's method is very simple and elegant when there is only one Skolem function to eliminate, but it becomes more complicated and difficult to handle when dealing with several Skolem functions. In fact, a nesting of the notion of forcing together with a technical result about elimination of definitions become necessary and the method looses some intuitive appeal. Instead, the use of Interactive realizability allows to deal with all the Skolem functions at the same time, and we conjecture that the resulting proofs are much shorter than the ones obtained by forcing. Moreover, the method of forcing seems a bit "magical" and it is much more natural to talk about states and approximations when dealing with programs.

Secondly, the theory of Interactive realizability offers a uniform explanation of a number of different phenomena. Rather than proving each particular meta-theoretic result about classical Arithmetic with an ad-hoc technique, one employs a single methodology. For example, one may prove conservativity of PA over HA for Π^0_2-formulas by a negative translation followed by Friedman's translation [12]; one may extract from proofs terms of Gödel's System T by realizability or functional interpretations [13]; one may prove the result about the elimination of Skolem functions with forcing; one may extract from proofs strategies in backtracking Tarski games by analyzing sequent calculus proofs [11]; one may obtain a simple ordinal analysis of $\mathsf{PA} + \mathsf{SK}$ by using update procedures [8]. Instead, with the theory of Interactive realizability one obtains all the results above as a consequence of a single concept (see [3, 5, 7]).

1.1 Plan of the paper

- In Section §2 we review the term calculus $\mathcal{T}_{\text{Class}}$ in which Interactive realizers are written, namely an extension of Gödel's system T plus Skolem function symbols for a countable collection of Skolem functions.
- In Section §3 we recall Interactive realizability, as described in [7], a computational semantics for $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$, an arithmetical system with functional variables which includes first-order classical Peano Arithmetic and Skolem axioms.
- In Section §4 we use Interactive realizability to prove the conservativity of $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$ over $\mathsf{HA}^\omega + \mathsf{EM}$ for arithmetical formulas.
- In Section §5 we explain in more detail how to formalize the proofs of Section 4 in $\mathsf{HA}^\omega + \mathsf{EM}$ and $\mathsf{HA} + \mathsf{EM}$.

2 The Term Calculus $\mathcal{T}_{\text{Class}}$

In this section we follow [7] and recall the typed lambda calculi $\mathcal{T}$ and $\mathcal{T}_{\text{Class}}$ in which interactive realizers are written. $\mathcal{T}$ is an extension of Gödel's system T (see Girard [14]) with some syntactic sugar. The basic objects of $\mathcal{T}$ are numerals, booleans, and its basic computational constructs are primitive recursion at all types, if-then-else, pairs, as in Gödel's T. $\mathcal{T}$ also includes as basic objects finite partial functions over $\mathbb{N}$ and simple primitive recursive operations over them. $\mathcal{T}_{\text{Class}}$ is obtained from $\mathcal{T}$ by adding on top of it a collection of Skolem function symbols $\Phi_0, \Phi_1, \Phi_2, \ldots$, of type $\mathtt{N} \rightarrow \mathtt{N}$, one for each arithmetical formula. The symbols are inert from the computational point of view and realizers are always computed with respect to some approximation of the Skolem maps represented by $\Phi_0, \Phi_1, \Phi_2, \ldots$.

2.1 Updates

In order to define $\mathcal{T}$, we start by introducing the concept of "update", which is nothing but a finite partial function over $\mathbb{N}$. Realizers of atomic formulas will return these finite partial functions, or "updates", as new pieces of information that they have learned about the Skolem function $\Phi_0, \Phi_1, \ldots$. Skolem functions, in turn, are used as "oracles" during computations in the system $\mathcal{T}_{\text{Class}}$. Updates are new associations input-output that are intended to correct, and in this sense, to *update*, wrong oracle values used in a computation.

■ **Definition 1** (Updates and Consistent Union). We define:

1. An update set U, shortly an *update*, is a finite set of triples of natural numbers representing a finite partial function from $\mathbb{N}^2$ to $\mathbb{N}$.

2. Two triples (a, n, m) and (a', n', m') of numbers are *consistent* if $a = a'$ and $n = n'$ implies $m = m'$. Two updates U_1, U_2 are consistent if $U_1 \cup U_2$ is an update.

3. $\mathbb{U}$ is the set of all updates.

4. The *consistent union* $U_1 \,\mathcal{U}\, U_2$ of $U_1, U_2 \in \mathbb{U}$ is $U_1 \cup U_2$ minus all triples of U_2 which are inconsistent with some triple of U_1.

The consistent union $U_1 \,\mathcal{U}\, U_2$ is an non-commutative operation: whenever a triple of U_1 and a triple of U_2 are inconsistent, we arbitrarily keep the triple of U_1 and we reject the triple of U_2, therefore for some U_1, U_2 we have $U_1 \,\mathcal{U}\, U_2 \neq U_2 \,\mathcal{U}\, U_1$. $\mathcal{U}$ represents a way of selecting a consistent subset of $U_1 \cup U_2$, such that $U_1 \mathcal{U} U_2 = \emptyset \implies U_1 = U_2 = \emptyset$.

2.2 The System $\mathcal{T}$

$\mathcal{T}$ is formally described in figure 1. Terms of the form $\mathsf{if}_A\, t_1\, t_2\, t_3$ will be sometimes written in the more legible form $\mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3$. A *numeral* is a term of the form $\mathsf{S}(\ldots\mathsf{S}(0)\ldots)$. For every update $U \in \mathbb{U}$, there is in $\mathcal{T}$ a constant $\overline{U} : \mathtt{U}$, where $\mathtt{U}$ is a new base type representing $\mathbb{U}$. We write $\varnothing$ for $\overline{\emptyset}$. In $\mathcal{T}$, there are four operations involving updates (see figure 1):

1. The first operation is denoted by the constant $\mathsf{min} : \mathtt{U} \rightarrow \mathtt{N}$. min takes as argument an update constant $\overline{U}$; it returns the minimum numeral a such that $(a, n, m) \in U$ for some $n, m \in \mathtt{N}$, if any exists; it returns 0 otherwise.

2. The second operation is denoted by the constant $\mathsf{get} : \mathtt{U} \to \mathtt{N}^3 \to \mathtt{N}$. get takes as arguments an update constant $\overline{U}$ and three numerals a, n, l; it returns m if $(a, n, m) \in U$ for some $m \in \mathtt{N}$ (i.e. if (a, n) belongs to the domain of the partial function U); it returns l otherwise.

3. The third operation is denoted by the constant $\mathsf{mkupd} : \mathtt{N}^3 \to \mathtt{U}$. mkupd takes as arguments three numerals a, n, m and transforms them into (the constant coding in $\mathcal{T}$) the update $\{(a, n, m)\}$.

4. The forth operation is denoted by the constant $\Cup : \mathtt{U}^2 \to \mathtt{U}$. $\Cup$ takes as arguments two update constants and returns the update constant denoting their consistent union.

We observe that the constants $\mathsf{min}, \mathsf{get}, \mathsf{mkupd}$ are just syntactic sugar and may be avoided by coding finite partial functions into natural numbers. System $\mathcal{T}$ may thus be coded in Gödel's T.

Types

$$\sigma, \tau ::= \mathtt{N} \mid \mathtt{Bool} \mid \mathtt{U} \mid \sigma \to \tau \mid \sigma \times \tau$$

Constants

$$c ::= \mathsf{R}_\tau \mid \mathsf{if}_\tau \mid 0 \mid \mathsf{S} \mid \mathtt{True} \mid \mathtt{False} \mid \mathsf{min} \mid \mathsf{get} \mid \mathsf{mkupd} \mid \Cup \mid \overline{U}\ (\forall U \in \mathbb{U})$$

Terms

$$t, u ::= \ c \mid x^\tau \mid tu \mid \lambda x^\tau u \mid \langle t, u\rangle \mid \pi_0 u \mid \pi_1 u$$

Typing Rules for Variables and Constants

$$x^\tau : \tau \mid 0 : \mathtt{N} \mid \mathsf{S} : \mathtt{N} \to \mathtt{N} \mid \mathtt{True} : \mathtt{Bool} \mid \mathtt{False} : \mathtt{Bool} \mid \overline{U} : \mathtt{U}\ (\text{for every } U \in \mathbb{U}) \mid \Cup : \mathtt{U} \to \mathtt{U} \to \mathtt{U}$$
$$\mid \mathsf{min} : \mathtt{U} \to \mathtt{N} \mid \mathsf{get} : \mathtt{U} \to \mathtt{N} \to \mathtt{N} \to \mathtt{N} \to \mathtt{N} \mid \mathsf{mkupd} : \mathtt{N} \to \mathtt{N} \to \mathtt{N} \to \mathtt{U}$$
$$\mid \mathsf{if}_\tau : \mathtt{Bool} \to \tau \to \tau \to \tau \mid \mathsf{R}_\tau : \tau \to (\mathtt{N} \to (\tau \to \tau)) \to \mathtt{N} \to \tau$$

Typing Rules for Composed Terms

$$\frac{t : \sigma \to \tau \qquad u : \sigma}{tu : \tau} \qquad \frac{u : \tau}{\lambda x^\sigma u : \sigma \to \tau} \qquad \frac{u : \sigma \qquad t : \tau}{\langle u, t\rangle : \sigma \times \tau} \qquad \frac{u : \tau_0 \times \tau_1}{\pi_i u : \tau_i}\ i \in \{0, 1\}$$

Reduction Rules All the usual reduction rules for simply typed lambda calculus (see Girard [14]) plus the rules for recursion, if-then-else and projections

$$\mathsf{R}_\tau uv0 \mapsto u \quad \mathsf{R}_\tau uv\mathsf{S}(t) \mapsto vt(\mathsf{R}_\tau uvt) \quad \mathsf{if}_\tau\, \mathtt{True}\, u\, v \mapsto u \quad \mathsf{if}_\tau\, \mathtt{False}\, u\, v \mapsto v \quad \pi_i\langle u_0, u_1\rangle \mapsto u_i, i = 0, 1$$

plus the following ones, assuming a, n, m, l be numerals:

$$\mathsf{min}\, \overline{U} \mapsto \begin{cases} a & \text{if } \exists m, n.\ (a, n, m) \in U \wedge \forall (b, i, j) \in U.\ a \leq b \\ 0 & \text{otherwise} \end{cases} \qquad \overline{U}_1 \Cup \overline{U}_2 \mapsto \overline{U_1 \mathcal{U} U_2}$$

$$\mathsf{get}\, \overline{U}\, a\, n\, l \mapsto \begin{cases} m & \text{if } \exists m.\ (a, n, m) \in U \\ l & \text{otherwise} \end{cases} \qquad \mathsf{mkupd}\, a\, n\, m \mapsto \overline{\{(a, n, m)\}}$$

Figure 1 the extension $\mathcal{T}$ of Gödel's system T

As proved in [1, 4], $\mathcal{T}$ is strongly normalizing, has the uniqueness-of-normal-form property and the following normal form theorem also holds.

Lemma 2 (Normal Form Property for $\mathcal{T} + C + \mathcal{R}$). *Assume that $\mathcal{R}$ is a functional set of reduction rules for C. Assume A is either an atomic type or a product type. Then any closed* *normal term $t \in \mathcal{T}$ of type A is: a numeral $n : \mathtt{N}$, or a boolean $\mathtt{True}, \mathtt{False} : \mathtt{Bool}$, or an update constant $\overline{U} : \mathtt{U}$, or a constant of type A, or a pair $\langle u, v\rangle : B \times C$.*

2.3 The System $\mathcal{T}_{\text{Class}}$

We now define a classical extension of $\mathcal{T}$, that we call $\mathcal{T}_{\text{Class}}$, with a Skolem function symbol for each arithmetical formula. The elements of $\mathcal{T}_{\text{Class}}$ will represent (non-computable) realizers.

■ **Definition 3** (The System $\mathcal{T}_{\text{Class}}$)**.** Define $\mathcal{T}_{\text{Class}} = \mathcal{T} + \mathcal{SC}$, where $\mathcal{SC}$ is a countable set of Skolem function constants, each one of type $\mathtt{N} \to \mathtt{N}$. We assume to have an enumeration $\Phi_0, \Phi_1, \Phi_2, \ldots$ of all the constants in $\mathcal{SC}$ (while generic elements of $\mathcal{SC}$ will be denoted with letters $\Phi, \Psi, \ldots$).

Every $\Phi \in \mathcal{SC}$ represents a *Skolem function* for some arithmetical formula $\exists y^{\mathtt{N}} A(x, y)$, taking as argument a number x and returning some y such that $A(x, y)$ is true if any exists, and an arbitrary value otherwise. In general, there is no set of computable reduction rules for the constants in $\mathcal{SC}$, and therefore no set of computable reduction rules for $\mathcal{T}_{\text{Class}}$. Each (in general, non-computable) term $t \in \mathcal{T}_{\text{Class}}$ is associated to a set $\{t[s] \mid s \in \mathcal{T}, s : \mathtt{N}^2 \to \mathtt{N}\} \subseteq \mathcal{T}$ of computable terms we call its "approximations", one for each term $s : \mathtt{N}^2 \to \mathtt{N}$ of $\mathcal{T}$, which is thought as a sequence $s_0, s_1, s_2, \ldots$ of computable approximations of the oracles $\Phi_0, \Phi_1, \Phi_2, \ldots$ (with s_i we denote $s(i)$).

■ **Definition 4** (Approximation at State)**.**

1. A *state* is a closed term of type $\mathtt{N}^2 \to \mathtt{N}$ of $\mathcal{T}$. If i is a numeral, with s_i we denote $s(i)$.

2. Assume $t \in \mathcal{T}_{\text{Class}}$ and s is a state. The "approximation of t at a state s" is the term $t[s]$ of $\mathcal{T}$ obtained from t by replacing each constant Φ_i with s_i.

3 Interactive Realizability for $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$

In this section we introduce a notion of realizability based on interactive learning for $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$, Heyting Arithmetic in all finite types (see e.g. Troelstra [25]) plus Excluded Middle and Skolem axiom schemes for all arithmetical formulas. Then we prove our main Theorem, the Adequacy Theorem: *"if a closed formula is provable in $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$, then it is realizable"*.

We first define the formal system $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$. We represent atomic predicates of $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$ with closed terms of $\mathcal{T}_{\text{Class}}$ of type $\mathtt{Bool}$. Terms of $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$ are elements of $\mathcal{T}_{\text{Class}}$ and thus may include the function symbols in $\mathcal{SC}$. We assume having in Gödel's T some terms $\Rightarrow_{\mathtt{Bool}}: \mathtt{Bool} \to \mathtt{Bool} \to \mathtt{Bool}, \neg_{\mathtt{Bool}} : \mathtt{Bool} \to \mathtt{Bool}, \vee_{\mathtt{Bool}} : \mathtt{Bool} \to \mathtt{Bool} \to \mathtt{Bool} \ldots$, implementing boolean connectives. As usual, we shall use infix notation: for example, we write $t_1 \Rightarrow_{\mathtt{Bool}} t_2$ in place of $\Rightarrow_{\mathtt{Bool}} t_1 t_2$ and similarly for the other connectives.

3.1 Language of $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$

We now define the language of the arithmetical theory $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$.

■ **Definition 5** (Language of $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$)**.** The language $\mathcal{L}_{\text{Class}}$ of $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$ is defined as follows.

1. The terms of $\mathcal{L}_{\text{Class}}$ are all $t \in \mathcal{T}_{\text{Class}}$.

2. The atomic formulas of $\mathcal{L}_{\text{Class}}$ are all $Q \in \mathcal{T}_{\text{Class}}$ such that $Q : \mathtt{Bool}$.

3. The formulas of $\mathcal{L}_{\text{Class}}$ are built from atomic formulas of $\mathcal{L}_{\text{Class}}$ by the connectives $\vee, \wedge, \rightarrow, \setminus, \forall, \exists$ as usual, with quantifiers possibly ranging over variables $x^\tau, y^\tau, z^\tau, \ldots$ of arbitrary finite type τ of $\mathcal{T}_{\text{Class}}$.

4. A formula of $\mathcal{L}_{\text{Class}}$ is said *arithmetical* if it does not contain constants in $\mathcal{SC}$ and all its quantifiers range over the type $\mathtt{N}$, i.e. it has one of the following forms: $\forall x^{\mathtt{N}} A, \exists x^{\mathtt{N}} A, A \vee B, A \wedge B, A \rightarrow B, A \setminus B, P$, with A, B arithmetical and P atomic formula of $\mathcal{T}$.

We denote with $\perp$ the atomic formula $\mathtt{False}$ and with $\neg A$ the formula $A \rightarrow \perp$. $A \setminus B$ is the dual of implication as in bi-intuitionistic logic and means "A and the opposite of B". If F is a formula of $\mathcal{L}_{\text{Class}}$ in the free variables $x_1^{\tau_1}, \ldots, x_n^{\tau_n}$ and $t_1 : \tau_1, \ldots, t_n : \tau_n$ are terms of $\mathcal{L}_{\text{Class}}$, with $F(t_1, \ldots, t_n)$ we shall denote the formula $F[t_1/x_1, \ldots, t_n/x_n]$. Sequences of variable $x_1^{\mathtt{N}}, \ldots, x_k^{\mathtt{N}}$ will be written as $\vec{x}$. We denote with $\langle \vec{x} \rangle$ a term of $\mathcal{T}$ in the free numeric variables $\vec{x}$ representing a injection of $\mathtt{N}^k$ into $\mathtt{N}$. Moreover, for every sequence of numerals $\vec{n} = n_1, \ldots, n_k$, we define $\langle \vec{n} \rangle := \langle \vec{x} \rangle [\vec{n}/\vec{x}]$ and assume that the function $\vec{n} \mapsto \langle \vec{n} \rangle$ is a bijection.

The *Excluded Middle axiom scheme* EM is defined as the set of all formulas of the form:

$$\forall \vec{x}^{\mathtt{N}}.\ A(\vec{x}) \vee \neg A(\vec{x})$$

where A is an arithmetical formula.

The *Skolem axiom scheme* SK contains for each arithmetical formula $A(\vec{x}, y)$ an axiom:

$$\forall \vec{x}^{\mathtt{N}}.\ \exists y^{\mathtt{N}} A(\vec{x}, y) \rightarrow A(\vec{x}, \Phi \langle \vec{x} \rangle)$$

with $\Phi \in \mathcal{SC}$. We assume that for every $\Phi \in \mathcal{SC}$ there is in SK one and only one formula in which Φ occurs. Such unique formula A is said to be the *formula associated to* Φ and Φ will be sometimes written as Φ_A. If s is a state and $\Phi_i = \Phi_A$, with s_A we denote s_i and with $\mathsf{mkupd}\, A\, u\, t$ we denote $\mathsf{mkupd}\, i\, u\, t$.

For each formula F of $\mathcal{L}_{\text{Class}}$, its involutive negation $F^\perp$ is defined by induction on F. First, we say that an atomic formula P is positive if it is of the form $\neg_{\mathtt{Bool}} \ldots \neg_{\mathtt{Bool}} Q$, Q is not of the form $\neg_{\mathtt{Bool}} R$, and the number of $\neg_{\mathtt{Bool}}$ in front of Q is even. Then we define:

$$\begin{array}{ll}
(\neg_{\mathtt{Bool}} P)^\perp = P \text{ (if } P \text{ positive)} & P^\perp = \neg_{\mathtt{Bool}} P \text{ (if } P \text{ positive)} \\
(A \wedge B)^\perp = A^\perp \vee B^\perp & (A \vee B)^\perp = A^\perp \wedge B^\perp \\
(A \rightarrow B)^\perp = A \setminus B & (A \setminus B)^\perp = A \rightarrow B \\
(\forall x^\tau A)^\perp = \exists x^\tau A^\perp & (\exists x^\tau A)^\perp = \forall x^\tau A^\perp
\end{array}$$

As usual, one has $(F^\perp)^\perp = F$.

Definition 6 (Set Γ). We fix an arbitrary finite set Γ of arithmetical formulas $A(\vec{x}, y)$ of $\mathcal{L}_{\text{Class}}$.

In the following, the set Γ will serve as a parameter in order to relativize the definitions of the realizability relation and of the ordering of states given in [7]. The idea is that we shall have to interpret proofs in the system $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$ and each one of them will use only a finite number of instances of EM and SK. Thus, realizers will contain only a finite number of Skolem functions, each one of them corresponding to some formula in Γ. The restriction of all the concepts to the set Γ is necessary in order to avoid to speak about the truth of an infinite number of formulas in the definitions we are going to give. In such a way, with a proper choice of Γ we shall be able to interpret any given proof of $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$.

3.2 Truth Value of a Formula in a State

The axioms of the system $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$ give a great computational power to the system $\mathcal{T}_{\text{Class}}$: one can "compute" by a term χ_F of $\mathcal{T}_{\text{Class}}$ the truth value of any arithmetical formula F. When one effectively evaluates χ_F in a particular state s, we say that one computes *the truth value of a formula F in a state s*.

■ **Definition 7** (Truth Value of a Formula F in a State s). For every arithmetical formula $F(\vec{x})$ of $\mathcal{L}_{\text{Class}}$ we define, by induction on F, a term χ_F : `Bool` of system $\mathcal{T}_{\text{Class}}$, with the same free variables of F:

$$\chi_P = P, \ P \text{ atomic}$$

$$\chi_{A\vee B} = \chi_A \vee_{\texttt{Bool}} \chi_B \qquad \chi_{\forall y^{\mathbb{N}} A} = \chi_A[\Phi_{A^\perp}\langle\vec{x}\rangle/y] \qquad \chi_{A \setminus B} = \chi_A \wedge_{\texttt{Bool}} \chi_{B^\perp}$$

$$\chi_{A\wedge B} = \chi_A \wedge_{\texttt{Bool}} \chi_B \qquad \chi_{\exists y^{\mathbb{N}} A} = \chi_A[\Phi_{A}\langle\vec{x}\rangle/y] \qquad \chi_{A \to B} = \chi_A \Rightarrow_{\texttt{Bool}} \chi_B$$

We define $F^s := \chi_F[s]$ and call it *the truth value of F in the state s.*

Intuitively, if $F(\vec{n})$ is a closed formula, our intended interpretation is:

1. $\chi_F(\vec{n})$ is a term of $\mathcal{T}_{\text{Class}}$ *denoting*, in any standard model of $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$, the truth value of $F(\vec{n})$.
2. $F^s(\vec{n})$ is a term of $\mathcal{T}$ *computing what would be* the truth value of $F(\vec{n})$ in some standard model of HA^ω+EM *under the (possible false) assumption* that the interpretation mapping Φ_i to s_i satisfies the axioms of SK.

We remark that thus $F^s(\vec{n})$ is only a *conditional* truth value: if $F^s(\vec{n})$ is not the correct truth value of $F(\vec{n})$ – it may well happen – then the interpretation mapping Φ_i in s_i does not satisfy the axioms of SK. This subtle point is what makes possible learning in Interactive realizability: whenever a contradiction follows, realizers are able to effectively find counter-examples to the assertion that the interpretation mapping Φ_i in s_i satisfies the axioms of SK. We also observe that this way of computing the truth of a formula comes from the epsilon substitution method (see Avigad [8], Mints et al. [19]).

Every state s is considered as an *approximation* of the Skolem functions denoted by the constants of $\mathcal{SC}$: for each formula A, s_A may be a correct approximation of Φ_A on some arguments, but wrong on other ones. More precisely, we are going to consider the set $\mathsf{def}(s)$ of the pairs $(i, \langle\vec{n}\rangle)$ such that $\Phi_i = \Phi_A$ and $A \in \Gamma \Rightarrow \exists y^{\mathbb{N}} A(\vec{n}, y) \to A(\vec{n}, s_i\langle\vec{n}\rangle)$ is true as the real "domain" of s, representing the set of arguments at which s_i is surely a correct approximation of Φ_i, in the sense that s_i returns an appropriate witness if any exists. We point out that if $\Phi_i = \Phi_A$ and $A \notin \Gamma$, then trivially $(i, \langle\vec{n}\rangle) \in \mathsf{def}(s)$. The choice is made just for technical convenience, since one is not interested in the behaviour of s outside Γ. We also define an ordering between states: we say that $s' \geq s$ if, intuitively, s' is at least as good an approximation as s. Thus, we ask that if s is a correct approximation at argument $(i, \langle\vec{n}\rangle)$ also s' is and in particular $s'_i\langle\vec{n}\rangle = s_i\langle\vec{n}\rangle$.

■ **Definition 8** (Domains, Ordering between States).

1. We define

$$\mathsf{def}(s) = \{(i, \langle\vec{n}\rangle) \mid \Phi_i = \Phi_A \text{ and } (A \in \Gamma \Rightarrow \exists y^{\mathbb{N}} A(\vec{n}, y) \to A(\vec{n}, s_i\langle\vec{n}\rangle))\}$$

where i and $\vec{n}$ range over numerals and sequences of numerals.

2. Let s and s' be two states. We define $s' \geq s$ if and only if for all $(i, \langle\vec{n}\rangle)$, $(i, \langle\vec{n}\rangle) \in \mathsf{def}(s)$ implies $s_i\langle\vec{n}\rangle = s'_i\langle\vec{n}\rangle$.

We remark that by definition, $s' \geq s$ implies $\mathsf{def}(s') \supseteq \mathsf{def}(s)$ and that thanks to the restriction to Γ the relation $s' \geq s$ is arithmetical, because the condition $(i, \langle\vec{n}\rangle) \in \mathsf{def}(s)$ is non-trivial only for finitely many i. From now onwards, for every pair of terms t_1, t_2 of system $\mathcal{T}$, we shall write $t_1 = t_2$ if they are the same term modulo the equality rules corresponding to the reduction rules of system $\mathcal{T}$ (equivalently, if they have the same normal form).

3.3 Interactive Realizability

For every formula A of $\mathcal{L}_{\text{Class}}$, we now define what type $|A|$ a realizer of A must have.

Definition 9 (Types for realizers). For each formula A of $\mathcal{L}_{\text{Class}}$ we define a type $|A|$ of $\mathcal{T}_{\text{Class}}$ by induction on A:

$$|P| = \mathtt{U},\ \text{if } P \text{ is atomic}$$

$$|A \wedge B| = |A| \times |B| \qquad |\exists x^\tau A| = \tau \times |A| \qquad |A \setminus B| = |A| \times |B^\perp|$$

$$|A \vee B| = \mathtt{Bool} \times (|A| \times |B|) \qquad |\forall x^\tau A| = \tau \to |A| \qquad |A \to B| = |A| \to |B|$$

Let now $\mathsf{p}_0 := \pi_0 : \sigma_0 \times (\sigma_1 \times \sigma_2) \to \sigma_0$, $\mathsf{p}_1 := \pi_0\pi_1 : \sigma_0 \times (\sigma_1 \times \sigma_2) \to \sigma_1$ and $\mathsf{p}_2 := \pi_1\pi_1 : \sigma_0 \times (\sigma_1 \times \sigma_2) \to \sigma_2$ be the three canonical projections from $\sigma_0 \times (\sigma_1 \times \sigma_2)$. We define the realizability relation $t \Vdash F$, where $t \in \mathcal{T}_{\text{Class}}$, $F \in \mathcal{L}_{\text{Class}}$ and $t : |F|$.

Definition 10 (Interactive Realizability). Assume s is a state, t is a closed term of $\mathcal{T}_{\text{Class}}$, $F \in \mathcal{L}_{\text{Class}}$ is a closed formula, and $t : |F|$. We define first the relation $t \Vdash_s F$ by induction and by cases according to the form of F:

1. $t \Vdash_s Q$ for some atomic Q if and only if $\overline{U} = t[s]$ implies:
   - for every $(i, \vec{n}, m) \in U$, $\Phi_i = \Phi_A$ for some $A \in \Gamma$, and $A^s(\vec{n}, s_i\langle\vec{n}\rangle) = \mathtt{False}$ and $A^s(\vec{n}, m) = \mathtt{True}$.
   - $\overline{U} = \varnothing$ implies $Q[s] = \mathtt{True}$

2. $t \Vdash_s A \wedge B$ if and only if $\pi_0 t \Vdash_s A$ and $\pi_1 t \Vdash_s B$

3. $t \Vdash_s A \vee B$ if and only if either $\mathsf{p}_0 t[s] = \mathtt{True}$ and $\mathsf{p}_1 t \Vdash_s A$, or $\mathsf{p}_0 t[s] = \mathtt{False}$ and $\mathsf{p}_2 t \Vdash_s B$

4. $t \Vdash_s A \to B$ if and only if for all u, if $u \Vdash_s A$, then $tu \Vdash_s B$

5. $t \Vdash_s A \setminus B$ if and only if $\pi_0 t \Vdash_s A$ and $\pi_1 t \Vdash_s B^\perp$

6. $t \Vdash_s \forall x^\tau A$ if and only if for all closed terms $u : \tau$ of $\mathcal{T}$, $tu \Vdash_s A[u/x]$

7. $t \Vdash_s \exists x^\tau A$ if and only for some closed term $u : \tau$ of $\mathcal{T}$, $\pi_0 t[s] = u$ and $\pi_1 t \Vdash_s A[u/x]$

We define $t \Vdash F$ if and only if for all states s of $\mathcal{T}$, $t \Vdash_s F$.

The ideas behind the definition of $\Vdash_s$ in the case of $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$ are those we already explained in [7]. A realizer is a term t of $\mathcal{T}_{\text{Class}}$, possibly containing some non-computable Skolem function of $\mathcal{SC}$; if such a function was computable, t would be an intuitionistic realizer. Since in general t is not computable, we calculate its approximation $t[s]$ at state s. t is an intelligent, self-correcting program, representing a proof/construction depending on the state s. The realizer *interacts* with the environment, which may provide a counter-proof, a counterexample invalidating the current construction of the realizer. But the realizer is always able to turn such a negative outcome into a positive information, which consists in some new piece of knowledge learned about some Skolem function Φ_i.

The next proposition tells that realizability at state s respects the notion of equality of $\mathcal{T}_{\text{Class}}$ terms, when the latter is relativized to state s. That is, if two terms are equal at the state s, then they realize the same formulas in the state s.

Proposition 11 (Saturation). *If $t_1[s] = t_2[s]$ and $u_1[s] = u_2[s]$, then $t_1 \Vdash_s B[u_1/x]$ if and only if $t_2 \Vdash_s B[u_2/x]$.*

Proof. By straightforward induction on A. ◀

In the following, we use a standard natural deduction system for $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$, together with a term assignment in the spirit of Curry-Howard correspondence for classical logic. We denote with $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK} \vdash t : A$ the derivability relation in that system, where t is a term of $\mathcal{T}_{\text{Class}}$ and A is a formula of $\mathcal{L}_{\text{Class}}$. All details can be found in [4], [7].

The main theorem about Interactive realizability is the Adequacy Theorem: if a closed formula is provable in $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$, then it is realizable (see [7] for a proof).

Theorem 12 (Adequacy Theorem). *If A is a closed formula such that $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK} \vdash t : A$ and all the subformulas of the instances of EM and SK used in the derivation belong to Γ, then $t \Vdash A$.*

4 Conservativity of $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$ over $\mathsf{HA}^\omega + \mathsf{EM}$ ($\mathsf{HA} + \mathsf{EM}$)

The aim of this section is to use Interactive realizability in order to prove that for every arithmetical formula A, if $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK} \vdash A$ then $\mathsf{HA}^\omega + \mathsf{EM} \vdash A$ ($\mathsf{HA} + \mathsf{EM} \vdash A$). Since we know by the Adequacy Theorem 12 that $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK} \vdash A$ implies $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK} \vdash t : A$ and HA^ω proves $t \Vdash A$, our goal is to show in $\mathsf{HA}^\omega + \mathsf{EM}$ that $t \Vdash A$ implies A.

The intuitive reason this latter result is true is the following: one can always find an approximation s of the Skolem functions of t which is good enough to contain all the information needed by t to compute the *true* witnesses for A against any particular purported counterexample. The idea is that one has only to collect finitely many values of each Skolem function called during the execution of the program represented by t. To this end, it suffices to invoke the excluded middle a number of times which, intuitively, can be expressed in a proof formalizable in $\mathsf{HA}^\omega + \mathsf{EM}$. This is possible because $\mathsf{HA}^\omega + \mathsf{EM}$ is strong enough to prove the normalization of each term t of $\mathcal{T}_{\text{Class}}$ with respect to any interpretation of its Skolem functions. Finally, if there existed a counterexample to A, it would be possible to falsify the construction of the realizer t in the state s. Since t is a self-correcting program, it would be able to correct one of the values of s it has used in the computation of some witness for A. But s is constructed as to be correct on all the values used by t, which entails a contradiction.

For example, let $A = \exists x^{\mathtt{N}} \forall y^{\mathtt{N}} \exists z^{\mathtt{N}} P(x, y, z)$. Then one can find a state s which contains all the values of the Skolem functions needed to compute $n = \pi_0 t[s]$. Suppose a counterexample m to the formula $\forall y^{\mathtt{N}} \exists z^{\mathtt{N}} P(n, y, z)$ existed. Then one can find a state $s' \geq s$ which contains all the values of the Skolem functions needed to compute $l = \pi_0\left((\pi_1 t)m\right)[s']$. Now, we would have that $P(n, m, l)$ is false; thus, $\pi_1\left((\pi_1 t)m\right)[s']$ would be equal to some update $\overline{U}$

containing some corrections to s'. We shall show that this will not be the case, and the intuitive reason is that s' can be chosen as to be correct everywhere it is needed.

We now elaborate our argument. We start with a definition axiomatizing the informal concept that a state s contains all the information needed to compute the normal form of a term t of ground type. Namely, if for every s' extending s the evaluation of t in the state s' gives the same result obtained evaluating t in s, then we may assume all the relevant information is already in s.

Definition 13 (Definition of a term in a state s). For every state s and term t of $\mathcal{T}_{\text{Class}}$ of atomic type, we define $t \downarrow^s$ (and we say "t is defined in s") as the statement: for all states $s' \geq s$, $t[s'] = t[s]$.

▶ Remark. There is another, perhaps more intuitive way to express the concept of "being defined in the state s". For every state s we may define a binary reduction relation $\overset{s}{\mapsto} \subseteq \mathcal{T}_{\text{Class}} \times \mathcal{T}_{\text{Class}}$ as follows: $t \overset{s}{\mapsto} u$ if either $t \mapsto u$ in $\mathcal{T}_{\text{Class}}$ or u is obtained from t by replacing one of its subterms $\Phi_i(n)$ with a numeral $m = s_i(n)$ such that $(i, n) \in \mathsf{def}(s)$. Then one could say that t is defined in s if $t \overset{s}{\mapsto} a$ where a is either a numeral, a boolean or an update. Though this approach works well, it is unsuitable to be directly formalized in HA^ω, because in that system one cannot express this syntactical reasoning on terms.

We now define for every type τ a set of "computable" terms of type τ by means of the usual Tait-style computability predicates [21]. In our case, following the approach of the previous discussion, we consider a term t of ground type to be computable if for every state s, one can find a state $s' \geq s$ such that t is defined in s'. The notion is lifted to higher types as usual.

Definition 14 (Computable terms).

For every type τ of $\mathcal{T}_{\text{Class}}$, we define a set of closed terms of $\mathcal{T}_{\text{Class}}$ of type τ as follows:

- $\|\mathtt{N}\|=\{t : \mathtt{N} \mid$ for all states s there is a state $s' \geq s$ such that $t \downarrow^{s'} \}$
- $\|\mathtt{Bool}\|=\{t : \mathtt{Bool} \mid$ for all states s there is a state $s' \geq s$ such that $t \downarrow^{s'} \}$
- $\|\mathtt{U}\|=\{t : \mathtt{U} \mid$ for all states s there is a state $s' \geq s$ such that $t \downarrow^{s'} \}$
- $\|\tau \to \sigma\|=\{t \mid \forall u \in \|\tau\|\, tu \in \|\sigma\|\}$
- $\|\tau \times \sigma\|=\{t \mid \pi_0 t \in \|\tau\|$and $\pi_1 t \in \|\sigma\|\}$

In order to show that every term t in $\mathcal{T}_{\text{Class}}$ is computable, as usual we need to prove that the set of computable terms is saturated with respect to some suitable relation. In our case, to terms are related if they are equal in all states greater than some state.

Lemma 15. *For every term $t : \rho$ of $\mathcal{T}_{\text{Class}}$, if for every state s there exists a state $s' \geq s$ and $u \in \|T\|$ such that for all state $s'' \geq s'$, $t[s''] = u[s'']$, then $t \in \|T\|$.*

Proof. By induction on the type ρ.

- $\rho = \mathtt{N}$. Let s be a state. We have to show that there exists a state $r \geq s$ such that $t \downarrow^r$. By assumption on t there exists a state $s' \geq s$ and $u \in \|\mathtt{N}\|$ such that for all $s'' \geq s'$, $t[s''] = u[s'']$. Since $u \in \|\mathtt{N}\|$, there exists $s'' \geq s'$ such that $u \downarrow^{s''}$. Let $r = s''$; we prove $t \downarrow^r$. Let $r' \geq r$. We have that $u[r'] = u[r]$, by $u \downarrow^r$, and $t[r'] = u[r']$, since $r' \geq s'$. Hence, $t[r'] = u[r] = t[r]$. We conclude $t \downarrow^r$ and finally $t \in \|\mathtt{N}\|$.

- $\rho = \mathtt{Bool}, \mathtt{U}$: as for the case $\rho = \mathtt{N}$.

- $\rho = \tau \to \sigma$. Let $v \in \|\tau\|$. We have to show that $tv \in \|\sigma\|$. Let s be any state. By assumption on t there exist a state $s' \geq s$ and $u \in \|\tau \to \sigma\|$ such that for all $s'' \geq s'$, $t[s''] = u[s'']$. Therefore for all $s'' \geq s'$, $tv[s''] = uv[s'']$ and $uv \in \|\sigma\|$. Hence, by induction hypothesis, $tv \in \|\sigma\|$.

- $\rho = \tau_0 \times \tau_1$. Let $i \in \{0,1\}$, we have to show that $\pi_i t \in \|\tau_i\|$. Let s be any state. By assumption on t there exist $s' \geq s$ and $u \in \|\tau_0 \times \tau_1\|$ such that for all $s'' \geq s'$, $t[s''] = u[s'']$. Therefore for all $s'' \geq s'$, $\pi_i t[s''] = \pi_i u[s'']$ and $\pi_i u \in \|\tau_i\|$. Hence, by induction hypothesis $\pi_i t \in \|\tau_i\|$.

◀

We are now ready to prove, by using the excluded middle alone, that every term t of $\mathcal{T}_{\text{Class}}$ is computable.

Theorem 16 (Computability Theorem).
Let $v : \tau$ be a term of $\mathcal{T}_{\text{Class}}$ and suppose that all the free variables of v are among $x_1^{\sigma_1}, \ldots, x_n^{\sigma_n}$. If $t_1 \in \|\sigma_1\|, \ldots, t_n \in \|\sigma_n\|$, then $v[t_1/x_1^{\sigma_1}, \ldots, t_n/x_n^{\sigma_n}] \in \|\tau\|$.

Proof. We proceed by induction on v. We first remark that if $u = t$ and $t \in \|\tau\|$, then $u \in \|\tau\|$ by trivial application of Lemma 15.

▶ Notation 1. For any term w in $\mathcal{T}_{\text{Class}}$, we denote $w[t_1/x_1^{\sigma_1}, \ldots, t_n/x^{\sigma_n}]$ with $\overline{w}$.

1. v is a variable $x_i^{\sigma_i} : \sigma_i$ and $\tau = \sigma_i$. Then, $\overline{v} = t_1 \in \|\sigma_i\| = \|\tau\|$.

2. v is 0, $\mathtt{True}$, $\mathtt{False}$, $\overline{U}$: trivial.

3. v is uw, then by means of typing rules, $u : \sigma \to \tau$, $w : \sigma$. Since by induction hypothesis $\overline{u} \in \|\sigma \to \tau\|$ and $\overline{w} \in \|\sigma\|$, we obtain $\overline{v} = \overline{uw} \in \|\tau\|$.

4. v is $\lambda x^{\tau_1}.u : \tau_1 \to \tau_2$. Then, by means of typing rules, $u : \tau_2$. Suppose now, for a term $t : \tau_1$ in $\mathcal{T}_{\text{Class}}$, that $t \in \|\tau_1\|$. We have to prove that $\overline{v}t \in \|\tau_2\|$. We have:

$$\begin{aligned}
\overline{v}t &= (\lambda x^{\tau_1}.u)[t_1/x_1^{\sigma_1} \cdots t_n/x_n^{\sigma_n}]t \\
&= (\lambda x^{\tau_1} u)t[t_1/x_1^{\sigma_1} \cdots t_n/x_n^{\sigma_n}] \\
&= u[t/x^{\tau_1}][t_1/x_1^{\sigma_1} \cdots t_n/x_n^{\sigma_n}] \\
&= u[t/x^{\tau_1}\, t_1/x_1^{\sigma_1} \cdots t_n/x_n^{\sigma_n}]
\end{aligned}$$

   By induction hypothesis, this latter term belongs to $\|\tau_2\|$. We conclude $\overline{v}t \in \|\tau_2\|$.

5. v is $\langle u, w\rangle : \tau_0 \times \tau_1$. By means of typing rules, $u : \tau_0$, $w : \tau_1$ and by induction hypothesis $\pi_0\overline{v} = \overline{u} \in \|\tau_0\|$ and $\pi_1\overline{v} = \overline{w} \in \|\tau_1\|$. The thesis $\overline{v} \in \|\tau_0 \times \tau_1\|$ follows by definition.

6. v is $\pi_i(u) : \tau_i$, $i = 0, 1$, where $u : \tau_0 \times \tau_1$. $\pi_i\overline{u} \in \|\tau_i\|$ because $\overline{u} \in \|\tau_0 \times \tau_1\|$ by induction hypothesis.

7. v is $\mathsf{if}_\tau : \mathtt{Bool} \to \tau \to \tau \to \tau$. Suppose that $u \in \|\mathtt{Bool}\|$, $u_1 \in \|\tau\|$, $u_2 \in \|\tau\|$. Then, for all states s there exists $s' \geq s$ such that $u \downarrow^{s'}$. We have to prove that $\mathsf{if}_\tau u u_1 u_2 \in \|\tau\|$. Let s be a state and let $s' \geq s$ be such that $u \downarrow^{s'}$. If $u[s'] = \mathtt{True}$, then for all $s'' \geq s'$, $\mathsf{if}_\tau u u_1 u_2[s''] = u_1[s'']$ and $u_1 \in \|\tau\|$. If $u[s'] = \mathtt{False}$, then for all $s'' \geq s'$, $\mathsf{if}_\tau u u_1 u_2[s''] = u_2[s'']$ and $u_2 \in \|\tau\|$. By Lemma 15, we conclude $\mathsf{if}_\tau u u_1 u_2 \in \|\tau\|$.

8. v is $\mathsf{R}_\tau : \tau \to (\mathtt{N} \to (\tau \to \tau)) \to \mathtt{N} \to \tau$. Suppose that $u \in \|\tau\|$, $w \in \|\mathtt{N} \to (\tau \to \tau)\|$, $z \in \|\mathtt{N}\|$. We have to prove that $\mathsf{R}_\tau\ uwz \in \|\tau\|$. By a plain induction, it is possible to prove, for each numeral n, $\mathsf{R}_\tau\ uwn \in \|\tau\|$. Let s be a state and let $s' \geq s$ be such that $z \downarrow^{s'}$. Let $z[s'] = n$ with n numeral. Then for all $s'' \geq s'$,

$$\mathsf{R}_\tau uvz[s''] = \mathsf{R}_\tau uvn[s''] \in \|\tau\|$$

By Lemma 15, we conclude $\mathsf{R}_\tau uwz \in \|\tau\|$.

9. v is $\mathsf{min} : \mathtt{U} \to \mathtt{N}$. Suppose, for a term u in $\mathcal{T}_{\text{Class}}$, that $u \in \|U\|$. Let s be a state. Since $u \in \|U\|$, there exists $s' \geq s$ such that $u \downarrow^{s'}$. We have to prove that $\mathsf{min}\ u \in \|\mathtt{N}\|$. There exists an update $\overline{U}$ such that for all $s'' \geq s'$, $u[s''] = \overline{U}$. Then for all $s'' \geq s'$, $\min u[s''] = \min \overline{U} = n$ for some numeral n. By definition of $\|\mathtt{N}\|$, $\mathsf{min}\ u \in \|\mathtt{N}\|$.

10. v is $\Cup : \mathtt{U} \to \mathtt{U} \to \mathtt{U}$. Suppose that $u_1 \in \|\mathtt{U}\|$ and $u_2 \in \|\mathtt{U}\|$. We have to prove that $\Cup\ u_1 u_2 \in \|\mathtt{U}\|$. Let s be a state. Since $u_1 \in \|\mathtt{U}\|$ there exists $s' \geq s$ such that $u_1 \downarrow^{s'}$. Since $u_2 \in \|\mathtt{U}\|$, there exists $s'' \geq s'$ such that $u_2 \downarrow^{s''}$. Therefore, there exist two constants $\overline{U}_1$ and $\overline{U}_2$ such that for all $s''' \geq s''$, $u_1[s'''] = \overline{U}_1$ and $u_2[s'''] = \overline{U}_2$. Finally, for all $s''' \geq s''$,

$$\Cup\ u_1 u_2[s'''] = \Cup\ \overline{U}_1 \overline{U}_2 = \overline{U}_3$$

and by definition of $\|\mathtt{U}\|$, $\Cup\ u_1 u_2 \in \|\mathtt{U}\|$.

11. v is S, mkupd or get. Analogous to the previous case.

12. v is a constant $\Phi_i : \mathtt{N} \to \mathtt{N}$ in $\mathcal{SC}$. Suppose now, for a term $u : \mathtt{N}$, that $u \in \|\mathtt{N}\|$. We have to prove that $\Phi_i u \in \|\mathtt{N}\|$. Let s be a state. We must show that there exists a $s' \geq s$ such that $\Phi_i u \downarrow^{s'}$. Since $u \in \|\mathtt{N}\|$, there exists a state $s' \geq s$ such that $u \downarrow^{s'}$. Let $n = u[s']$, with n numeral, and $m = s'_i(n)$. Let $\Phi_i = \Phi_{A(x,y)}$. If $A \notin \Gamma$, then trivially $(i, n) \in \mathsf{def}(s')$ by definition 8. Therefore for all $s'' \geq s'$, $\Phi_i u[s''] = s''_i(n) = m$ and we are done. Hence, we may assume $A \in \Gamma$. There are two cases, and this is the only point of this proof in which we use EM.

a. $A(n, m)$ is true. Therefore, for all $s'' \geq s'$, $s''_i(n) = m$ because $(i, n) \in \mathsf{def}(s')$. Thus, for all $s'' \geq s'$, $\Phi_i u[s''] = s''_i(n) = m$, which is the thesis.

b. $A(n, m)$ is false. If there exists l such that $A(n, l)$ is true, then let

$$s'' := \lambda x^{\mathtt{N}} \lambda y^{\mathtt{N}}.\, \mathsf{if}\ x = i \wedge_{\mathtt{Bool}} y = n\ \mathsf{then}\ m\ \mathsf{else}\ s'_x(y)$$

Then, for all $s''' \geq s''$, $s'''_i(n) = l$ because $(i, l) \in \mathsf{def}(s'')$. Thus, for all $s''' \geq s''$, $\Phi_i u[s'''] = s'''_i(n) = l$, which is the thesis. If there is no l such that $A(n, l)$ is true, then trivially $(i, n) \in \mathsf{def}(s')$. Thus for all $s'' \geq s'$, $\Phi_i u[s''] = s''_i(n) = m$ and we are done.

◀

According to the Definition 7 of the truth value A^s of a formula A in a state s, when we compute A^s we need only a finite number of Skolem function values, one for each quantifier of A. Thus, we can show with the excluded middle that for every state s there exists a state $s' \geq s$ such that when we evaluate A in the state s' we obtain the real truth value of A.

■ **Proposition 17.** *Let $A(\vec{x})$ be any arithmetical formula and $\vec{n}$ be numerals. For every state s, there exists a state $s' \geq s$ such that $A^{s'}(\vec{n}) = \texttt{True}$ if and only if $A(\vec{n})$ is true.*

Proof. We prove the thesis by induction on A. Let s be any state. The cases in which A is atomic or $A = B \vee C, B \wedge C, B \to C$ are trivial. Let us consider those in which A starts with a quantifier.

- $A(\vec{n}) = \exists y^{\mathtt{N}} B(\vec{n}, y)$. By the excluded middle, we extend s to a state $s' \geq s$ such that $m = s'_B\langle\vec{n}\rangle$ implies that
$$\exists y^{\mathtt{N}} B(\vec{n}, y) \to B(\vec{n}, m)$$
By induction hypothesis, there exists a state $s'' \geq s'$ such that $B(\vec{n}, m)$ is true if and only if
$$B^{s''}(\vec{n}, m) = \chi_B(\vec{n}, m)[s''] = \texttt{True}$$
Assuming $\Phi_i = \Phi_B$, since $(i, \langle\vec{n}\rangle) \in \mathsf{def}(s')$, we have $s''_B\langle\vec{n}\rangle = s'_B\langle\vec{n}\rangle$. Since
$$A^{s''}(\vec{n}) = \chi_B(\vec{n}, \Phi_B\langle\vec{n}\rangle)[s''] = \chi_B(\vec{n}, m)[s'']$$
and $A(\vec{n})$ is equivalent to $B(\vec{n}, m)$, we obtain the thesis.

- $A(\vec{n}) = \forall y^{\mathtt{N}} B(\vec{n}, y)$. By the excluded middle, we extend s to a state $s' \geq s$ such that $m = s'_{B^\perp}\langle\vec{n}\rangle$ implies that
$$\exists y^{\mathtt{N}} B^\perp(\vec{n}, y) \to B^\perp(\vec{n}, m)$$
By induction hypothesis, there exists a state $s'' \geq s'$ such that $B^\perp(\vec{n}, m)$ is true if and only if
$$(B^\perp)^{s''}(\vec{n}, m) = \chi_{B^\perp}[s''](\vec{n}, m) = \texttt{True}$$
Assuming $\Phi_i = \Phi_{B^\perp}$, since $(i, \langle\vec{n}\rangle) \in \mathsf{def}(s')$, we have $s''_{B^\perp}\langle\vec{n}\rangle = s'_{B^\perp}\langle\vec{n}\rangle$. Since
$$A^{s''}(\vec{n}) = \chi_{B^\perp}(\vec{n}, \Phi_{B^\perp}\langle\vec{n}\rangle)[s''] = \chi_{B^\perp}(\vec{n}, m)[s'']$$
we obtain the thesis.

◀

Now we prove a special case of the statement that the realizability of a formula implies the formula itself. Namely, we show that t realizes $\perp$ implies $\perp$. The idea, as we have explained before, is to find a state s which contains all the information needed to evaluate t.

■ **Theorem 18** (Consistency of Interactive Realizability). *For every closed term t of $\mathcal{T}_{\mathit{Class}}$, $t \not\Vdash \perp$. In particular, for every state s, there exists a state $s' \geq s$ such that $t \not\Vdash_{s'} \perp$.*

Proof. Suppose, for the sake of contradiction, that there exists a term t such that $t \Vdash \bot$. Let s be any state. Since $t : \mathtt{U}$, by theorem 16 we have $t \in \|\mathtt{U}\|$ and therefore there exists a state $r \geq s$ such that $t \downarrow^r$. Let $t[r] = \overline{U}$ for some update U. Since $t \Vdash_r \bot$, U is non-empty: let $(i, \vec{n}, m) \in U$. By application of theorem 16, if $\Phi_i = \Phi_A$, there exists a state $q \geq r$ such that $\chi_A(\vec{n}, m) \downarrow^q$. By definition,

$$A^q(\vec{n}, m) = \chi_A(\vec{n}, m)[q] = b$$

for some boolean b. Since $t \Vdash_q \bot$ and $t[q] = \overline{U}$ (because $t \downarrow^r$ and $q \geq r$), we obtain by definition of realizability that $b = \mathtt{True}$. Let $q_i\langle\vec{n}\rangle = l$. We have two possibilities:

1. $A(\vec{n}, l)$ is false. We define the state

$$s' := \lambda x^{\mathtt{N}} \lambda y^{\mathtt{N}}.\, \mathsf{if}\ x = i \wedge_{\mathtt{Bool}} y = \langle\vec{n}\rangle\ \mathsf{then}\ m\ \mathsf{else}\ q_x(y)$$

Then, $s' \geq q$, for $A(\vec{n}, l)$ is false. Moreover, since $\chi_A(\vec{n}, m) \downarrow^q$, for all $q' \geq q$, $\chi_A(\vec{n}, m)[q'] = b$; by Proposition 17, there exists $q' \geq q$, such that $\chi_A(\vec{n}, m)[q'] = \mathtt{True}$ if and only if $A(\vec{n}, m)$ is true. Since $\chi_A(\vec{n}, m)[q'] = b = \mathtt{True}$, we have that $A(\vec{n}, m)$ is true. By assumption on t, we have $t \Vdash_{s'} \bot$ and $t[s'] = \overline{U}$, because $s' \geq r$. Since $s'_i\langle\vec{n}\rangle = m$, by definition of $t \Vdash_{s'} \bot$ we would have both $A^{s'}(\vec{n}, m) = \mathtt{False}$ and $A^{s'}(\vec{n}, m) = \mathtt{True}$, which is a contradiction.

2. $A(\vec{n}, l)$ is true. By Proposition 17, there is a state $s' \geq q$ such that $A^{s'}(\vec{n}, l) = \mathtt{True}$. By assumption on t, we have $t \Vdash_{s'} \bot$ and $t[s'] = \overline{U}$. But $q_i\langle\vec{n}\rangle = l$, $A(\vec{n}, l)$ is true and $s' \geq q$; therefore $(i, \vec{n}) \in \mathsf{def}(q)$ and hence $s'_i\langle\vec{n}\rangle = l$. By definition of $t \Vdash_{s'} \bot$, we would have $A^{s'}(\vec{n}, l) = \mathtt{False}$ and $A^{s'}(\vec{n}, m) = \mathtt{True}$, which is in contradiction with $A^{s'}(\vec{n}, l) = \mathtt{True}$.

◀

Finally, we are in a position to prove in $\mathsf{HA}^\omega + \mathsf{EM}$ that the realizability of a formula A implies its truth.

■ **Theorem 19** (Soundness of Realizability). *Let A be any $\to$-free arithmetical formula and suppose $t \Vdash A$. Then A is true.*

Proof. We prove a stronger statement. Let s be a state and suppose that for all $s' \geq s$, $t \Vdash_{s'} A$. We prove by induction on A, that A is true.

- $A = P$, with P atomic. Suppose, by the way of contradiction, that P is false. Then we have that for all $s' \geq s$, $t \Vdash_{s'} \bot$, which is impossible by Theorem 18.

- $A = B \wedge C$. Then, for all $s' \geq s$, $t \Vdash_{s'} A$ and $t \Vdash_{s'} B$. By induction hypothesis A and B are true, and we obtain the thesis.

- $A = B \vee C$. By Theorem 16, there exists a state $r \geq s$ such that $\mathsf{p}_0 t \downarrow^r$. Let $\mathsf{p}_0 t[r] = b$ with b boolean, say $b = \mathtt{True}$. Then, by defintion, for every $r' \geq r$, $\mathsf{p}_0 t[r'] = \mathtt{True}$ and therefore $t \Vdash_{r'} A$. By induction hypothesis A is true, and we obtain the thesis.

- $A = \forall x^{\mathtt{N}} B$. Let n be any numeral. Then, for all $s' \geq s$, $tn \Vdash_{s'} B(n)$. By induction hypothesis $B(n)$ is true. Therefore, $\forall x^{\mathtt{N}} B$ is true, and we obtain the thesis.

- $A = \exists x^{\mathtt{N}} B$. By Theorem 16, there exists a state $r \geq s$ such that $\pi_0 t \downarrow^r$. Let $\pi_0 t[r] = n$ with n numeral. Then, by definition, for every $r' \geq r$, $\pi_0 t[r'] = n$ and therefore $t \Vdash_{r'} B(n)$. By induction hypothesis $B(n)$ is true, and we obtain the thesis.

◀

Since all the proofs given in this section are formalizable in $\mathsf{HA}^\omega + \mathsf{EM}$ (see Section 5), we are able to prove the conservativity of $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$ over $\mathsf{HA}^\omega + \mathsf{EM}$ for arithmetical formulas.

Theorem 20 (Conservativity of $\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK}$ over $\mathsf{HA}^\omega + \mathsf{EM}$)**.** *Let A be a closed arithmetical formula, and suppose*

$$\mathsf{HA}^\omega + \mathsf{EM} + \mathsf{SK} \vdash A$$

Then:

$$\mathsf{HA}^\omega + \mathsf{EM} \vdash A \tag{1}$$

$$\mathsf{HA} + \mathsf{EM} \vdash A \tag{2}$$

Proof.

1. We may assume that A is $\rightarrow$-free. Otherwise,

$$\mathsf{HA}^\omega + \mathsf{EM} \vdash A \leftrightarrow B$$

with B $\rightarrow$-free and we consider B. Since Γ is arbitrary, we may assume that all the subformulas of the instances of EM and SK used in the derivation belong to Γ. By formalization of the Adequacy Theorem 12 in HA^ω (see Section 5), we obtain that $\mathsf{HA}^\omega \vdash t \Vdash A$ for some term t of $\mathcal{T}_{\text{Class}}$. By formalization of the proof of Theorem 19 in $\mathsf{HA}^\omega + \mathsf{EM}$, we obtain that $\mathsf{HA}^\omega + \mathsf{EM} \vdash (t \Vdash A) \rightarrow A$. We conclude $\mathsf{HA}^\omega + \mathsf{EM} \vdash A$.

2. There are at least two ways to obtain the thesis. On one hand, we may use (1) and the standard result about the conservativity of $\mathsf{HA}^\omega + \mathsf{EM}$ over $\mathsf{HA} + \mathsf{EM}$ for arithmetical formulas (see for example Troesltra [23]). On the other hand, we may code directly terms of system $\mathcal{T}_{\text{Class}}$ into natural numbers and then express the proofs of point 1) in $\mathsf{HA} + \mathsf{EM}$ (see Section 5).

◀

5 Formalization of the Proofs in PA and in $\mathsf{HA}^\omega + \mathsf{EM}$

In this section we explain how to formalize in PA and $\mathsf{HA}^\omega + \mathsf{EM}$ the proof of the Adequacy Theorem 12 of Section 3, the proofs of the Computability Theorem 16 and the Soundness Theorem 19 of Section 4. We start with the case of PA.

5.1 Formalization in PA

One can routinely code in PA all the concepts we have so far used. As in Tait [21], one may code the terms of $\mathcal{T}_{\text{Class}}$ with natural numbers and successively the definition of the realizability and computability predicates with arithmetical formulas. Since neither set-theoretic concepts nor Skolem axioms are employed in any of the given proofs, everything can be coded in PA.

5.2 Formalization in $\mathsf{HA}^\omega + \mathsf{EM}$

Instead of coding everything into natural numbers, which is of limited practical interest, it is more satisfying to formalize our proofs directly in $\mathsf{HA}^\omega + \mathsf{EM}$. There is no serious obstacle to this end, except for a small formalization issue: the notion $t[s]$ of evaluation of a term t of $\mathcal{T}_{\text{Class}}$ in a state s, which we have heavily used in the definitions of the realizability and computability predicates, is not directly representable in $\mathsf{HA}^\omega + \mathsf{EM}$. To begin with, terms of $\mathcal{T}_{\text{Class}}$ may contain some constant $\Phi \in \mathcal{SC}$ which does not belong to the language of HA^ω. This problem is easily solved by considering terms of the form $t[s]$ with s state variable. However, in the definition of Interactive realizability for implication and in the statement of the Computability Theorem one needs to define formulas $x \Vdash A$ and $x \in \|\mathtt{N}\|$, where x is a variable. In these definitions it is necessary to speak about the substitution of an actual state s in the body of a variable x, which is impossible in HA^ω (remember that x represents a term $t[s]$ of $\mathcal{T}$). This last issue is overcome quite easily by considering in place of a term $t : \tau$ in $\mathcal{T}_{\text{Class}}$ the term $\lambda s^{\mathtt{S}}.t[s] : \mathtt{S} \to \tau$, where $\mathtt{S} := \mathtt{N}^2 \to \mathtt{N}$ is the type of states. In this way, one makes explicit the functional dependence of t from the state s and transforms t into an object having a semantical denotation. It is however necessary to slightly adapt the definitions of realizability and computability, which is what we are going to do.

First, we give an alternative definition of Interactive realizability, which is shown in [4] to be equivalent to Kreisel's modified realizability for HA^ω applied to some Friedman translation of formulas. We denote with $\mathcal{L}$ the restriction of the language $\mathcal{L}_{\text{Class}}$ to the formulas not containing any Skolem function constant $\Phi \in \mathcal{SC}$.

Definition 21 (Alternative Definition of Interactive Realizability). Assume $s : \mathtt{S}$ is a closed term of $\mathcal{T}$, t is a closed term of $\mathcal{T}$, $D \in \mathcal{L}$ is a closed formula of $\mathcal{L}$, and $t : |D|$. We define by induction on D the relation $t \Vdash_s D$:

1. $t \Vdash_s Q$ if and only if $t = \overline{U}$ implies:
   - for every $(i, \vec{n}, m) \in U$, $\Phi_i = \Phi_A$ for some $A \in \Gamma$, and $A^s(\vec{n}, s_i\langle\vec{n}\rangle) = \mathtt{False}$ and $A^s(\vec{n}, m) = \mathtt{True}$.
   - $U = \varnothing$ implies $Q = \mathtt{True}$
2. $t \Vdash_s A \wedge B$ if and only if $\pi_0 t \Vdash_s A$ and $\pi_1 t \Vdash_s B$
3. $t \Vdash_s A \vee B$ if and only if either $\mathsf{p}_0 t = \mathtt{True}$ and $\mathsf{p}_1 t$ mr A, or $\pi_0 t = \mathtt{False}$ and $\mathsf{p}_1 t$ mr B
4. $t \Vdash_s A \to B$ if and only if for all u, if $u \Vdash_s A$, then $tu \Vdash_s B$
5. $t \Vdash_s \forall x^\tau A$ if and only if for all closed terms $u : \tau$ of $\mathcal{T}$, $tu \Vdash_s A[u/x]$
6. $t \Vdash_s \exists x^\tau A$ if and only for some closed term $u : \tau$ of $\mathcal{T}$, $\pi_0 t = u$ and $\pi_1 t \Vdash_s A[u/x]$

One can prove straightforwardly, as in [4], that our first Definition 10 of Interactive realizability is equivalent to this alternative one.

Theorem 22 (Characterization of Interactive Realizability). *Let $t \in \mathcal{T}_{Class}$ and s be a state. Then, for every $B \in \mathcal{L}_{Class}$*

$$t \Vdash_s B \iff t[s] \Vdash_s B[s]$$

Theorem 22 allows us to replace in our conservativity proof the expression $t \Vdash A$ with the expression $\forall s^{\mathsf{S}}.\, t[s] \Vdash_s A[s]$, which is a formula of HA^ω. Moreover, the Adequacy Theorem for $\Vdash$ is formalizable in HA^ω, since it is a special case of the Adequacy Theorem for modified realizability, which is formalizable in that system (see [24]).

Secondly, we adapt the notion of computability to terms of type $\mathtt{S} \to \tau$. For every pair of terms $t, u \in \mathcal{T}$ respectively of type $\mathtt{S} \to (\sigma \to \tau)$ and $\mathtt{S} \to \sigma$, we define the following notion of application:

$$t \cdot u := \lambda s^{\mathtt{S}}.ts(us)$$

For every term $t \in \mathcal{T}$ of type $\mathtt{S} \to (\tau_0 \times \tau_1)$ and for $i = 0, 1$, we define the following notion of projection:

$$\pi_i t := \lambda s^{\mathtt{S}}.\pi_i ts$$

Finally, for every constant term $\mathsf{c} \notin \mathcal{SC}$, we define $\mathsf{c}^* := \lambda s^{\mathtt{S}}\mathsf{c}$. We now adapt Definition 13 and Definition 14. Since there is no possibility of confusion, we maintain the same notations of Section 4 but with the new specified meaning.

Definition 23 (Definition of a term in a state s)**.** For every state s and term $t : \mathtt{S} \to \tau$ of $\mathcal{T}$ with τ atomic type, we define $t \downarrow^s$ (and we say "t is defined in s") as the statement: for all states $s' \geq s$, $ts' = ts$.

Definition 24 (Computable terms)**.**
For every type τ of $\mathcal{T}$, we define a set of closed terms of $\mathcal{T}$ of type $\mathtt{S} \to \tau$ as follows:

- $\|\mathtt{N}\|=\{t : \mathtt{S} \to \mathtt{N} \mid$ for all states s there is a state $s' \geq s$ such that $t \downarrow^{s'}$ $\}$
- $\|\mathtt{Bool}\|=\{t : \mathtt{S} \to \mathtt{Bool} \mid$ for all states s there is a state $s' \geq s$ such that $t \downarrow^{s'}$ $\}$
- $\|\mathtt{U}\|=\{t : \mathtt{S} \to \mathtt{U} \mid$ for all states s there is a state $s' \geq s$ such that $t \downarrow^{s'}$ $\}$
- $\|\tau \to \sigma\|=\{t \mid \forall u \in \|\tau\| \;\; t \cdot u \in \|\sigma\|\}$
- $\|\tau \times \sigma\|=\{t \mid \pi_0 t \in \|\tau\|$and $\pi_1 t \in \|\sigma\|\}$

The proof of Lemma 15 is easily adapted.

Lemma 25. *For every term $t : \mathtt{S} \to \rho$ of $\mathcal{T}$, if for every state s there exists a state $s' \geq s$ and $u \in \|\rho\|$ such that for all states $s'' \geq s'$, $ts'' = us''$, then $t \in \|\rho\|$.*

Proof. By induction on the type ρ.

- $\rho = \mathtt{N}$. Let s be a state. We have to show that there exists a state $r \geq s$ such that $t \downarrow^r$. By assumption on t there exists a state $s' \geq s$ and $u \in \|\mathtt{N}\|$ such that for all $s'' \geq s'$, $ts'' = us''$. Thus, there exists $s'' \geq s'$ such that $u \downarrow^{s''}$. Let $r = s''$; we prove $t \downarrow^r$. Let $r' \geq r$. We have that $ur' = ur$, by $u \downarrow^r$, and $tr' = ur'$, since $r' \geq s'$. Hence, $tr' = ur = tr$. We conclude $t \downarrow^r$ and finally $t \in \|\rho\|$.
- $\rho = \mathtt{Bool}, \mathtt{U}$: as for the case $\rho = \mathtt{N}$.
- $\rho = \tau \to \sigma$. Let $v \in \|\tau\|$. We have to show that $t \cdot v \in \|\sigma\|$. Let s be any state. By assumption on t there exists a state $s' \geq s$ and $u \in \|\tau \to \sigma\|$ such that for all $s'' \geq s'$, $ts'' = us''$. Therefore for all $s'' \geq s'$,

$$(t \cdot v)s'' = ts''(vs'') = us''(vs'') = (u \cdot v)s''$$

and $u \cdot v \in \|\sigma\|$. Hence, by induction hypothesis, $t \cdot v \in \|\sigma\|$.

- $\rho = \tau_0 \times \tau_1$. Let $i \in \{0,1\}$, we have to show that $\pi_i t \in \|\tau_i\|$. Let s be any state. By assumption on t there exists $s' \geq s$ and $u \in \|\tau_0 \times \tau_1\|$ such that for all $s'' \geq s'$, $ts'' = us''$. Therefore for all $s'' \geq s'$,

$$(\pi_i t)s'' = \pi_i(ts'') = \pi_i(us'') = (\pi_i u)s''$$

and $\pi_i u \in \|\tau_i\|$. Hence, by induction hypothesis $\pi_i t \in \|\tau_i\|$.

◀

Also the proof of the Computability Theorem remains substantially the same.

Theorem 26 (Computability Theorem).
Let $v : \tau$ be a term of $\mathcal{T}_{\text{Class}}$ and suppose that all the free variables of v are among $x_1^{\sigma_1}, \ldots, x_n^{\sigma_n}$. If $t_1 \in \|\sigma_1\|, \ldots, t_n \in \|\sigma_n\|$, then $\lambda s^{\mathtt{S}}.v[s][t_1 s/x_1^{\sigma_1}, \ldots, t_n s/x_n^{\sigma_n}] \in \|\tau\|$.

Proof. We proceed by induction on v.

▶ Notation 2. For any term w in $\mathcal{T}_{\text{Class}}$, we denote $\lambda s^{\mathtt{S}}.w[s][t_1 s/x_1^{\sigma_1}, \ldots, t_n s/x^{\sigma_n}]$ with $\overline{w}$.

1. v is a variable $x_i^{\sigma_i} : \sigma_i$ and $\tau = \sigma_i$. So $\overline{v} = \lambda s^{\mathtt{S}}.t_i s$. Since for all states s, $\overline{v}s = t_i s$ and $t_i \in \|\sigma_i\|$, by Lemma 25, $\lambda s^{\mathtt{S}}.t_i s \in \|\sigma_i\|$.

2. v is 0, $\mathtt{True}$, $\mathtt{False}$, $\overline{U}$: trivial.

3. v is uw, then by means of typing rules, $u : \sigma \to \tau$, $w : \sigma$. Since by induction hypothesis $\overline{u} \in \|\sigma \to \tau\|$ and $\overline{w} \in \|\sigma\|$, we obtain $\overline{u} \cdot \overline{w} \in \|\tau\|$. Moreover,

$$\overline{v} = \lambda s^{\mathtt{S}}.\overline{u}s(\overline{w}s) = \overline{u} \cdot \overline{w} \in \|\tau\|$$

By Lemma 25, we obtain $\overline{v} \in \|\tau\|$.

4. v is $\lambda x^{\tau_1}.u : \tau_1 \to \tau_2$. Then, by means of typing rules, $u : \tau_2$. Suppose now, for a term $t : \mathtt{S} \to \tau_1$ in $\mathcal{T}$, that $t \in \|\tau_1\|$. We have to prove that $\overline{v} \cdot t \in \|\tau_2\|$. We have:

$$\begin{aligned}
\overline{v} \cdot t &= \big(\lambda s^{\mathtt{S}}.(\lambda x^{\tau_1}.u[s])[t_1 s/x_1^{\sigma_1} \cdots t_n s/x_n^{\sigma_n}]\big) \cdot t \\
&= \lambda s^{\mathtt{S}}.(\lambda x^{\tau_1}.u[s])[t_1 s/x_1^{\sigma_1} \cdots t_n s/x_n^{\sigma_n}](ts) \\
&= \lambda s^{\mathtt{S}}.(\lambda x^{\tau_1} u[s])(ts)[t_1 s/x_1^{\sigma_1} \cdots t_n s/x_n^{\sigma_n}] \\
&= \lambda s^{\mathtt{S}}.u[s][ts/x^{\tau_1}][t_1 s/x_1^{\sigma_1} \cdots t_n s/x_n^{\sigma_n}]
\end{aligned}$$

By induction hypothesis, this latter term is in $\|\tau_2\|$. By Lemma 25 we conclude $\overline{v} \cdot t \in \|\tau_2\|$.

5. v is $\langle u, w \rangle : \tau_0 \times \tau_1$. By means of typing rules, $u : \tau_0$, $w : \tau_1$ and by induction hypothesis $\pi_0 \overline{v} = \lambda s^{\mathtt{S}}.\pi_0(\overline{v}s) = \overline{u} \in \|\tau_0\|$ and $\pi_1 \overline{v} = \lambda s^{\mathtt{S}}.\pi_1(\overline{v}s) = \overline{w} \in \|\tau_1\|$. The thesis $\overline{v} \in \|\tau_0 \times \tau_1\|$ follows by definition.

6. v is $\pi_i(u) : \tau_i$, $i \in \{0,1\}$, where $u : \tau_0 \times \tau_1$. Then $\pi_i \overline{u} \in \|\tau_i\|$ because $\overline{u} \in \|\tau_0 \times \tau_1\|$ by induction hypothesis. Moreover,

$$\overline{v} = \lambda s^{\mathtt{S}}.\pi_i(\overline{u}s) = \pi_i \overline{u}$$

By Lemma 25, we obtain $\overline{v} \in \|\tau_i\|$.

7. v is $\mathsf{if}_\tau : \mathtt{Bool} \to \tau \to \tau \to \tau$. Suppose that $u \in \|\mathtt{Bool}\|$, $u_1 \in \|\tau\|$, $u_2 \in \|\tau\|$. Then, for all states s there exists $s' \geq s$ such that $u \downarrow^{s'}$. We have to prove that $\mathsf{if}^*_\tau \cdot u \cdot u_1 \cdot u_2 \in \|\tau\|$. Let s be a state and let $s' \geq s$ be such that $u \downarrow^{s'}$. If $us' = \mathtt{True}$, then for all $s'' \geq s'$,

$$(\mathsf{if}^*_\tau \cdot u \cdot u_1 \cdot u_2)\, s'' = \mathsf{if}_\tau(us'')(u_1 s'')(u_2 s'') = u_1 s''$$

and $u_1 \in \|\tau\|$. If $us' = \mathtt{False}$, then for all $s'' \geq s'$,

$$(\mathsf{if}^*_\tau \cdot u \cdot u_1 \cdot u_2)\, s'' = \mathsf{if}_\tau(us'')(u_1 s'')(u_2 s'') = u_2 s''$$

and $u_2 \in \|\tau\|$. By Lemma 25, we conclude $\mathsf{if}^*_\tau \cdot u \cdot u_1 \cdot u_2 \in \|\tau\|$.

8. v is $\mathsf{R}_\tau : \tau \to (\mathtt{N} \to (\tau \to \tau)) \to \mathtt{N} \to \tau$. Suppose that $u \in \|\tau\|$, $w \in \|\mathtt{N} \to (\tau \to \tau)\|$, $z \in \|\mathtt{N}\|$. We have to prove that $\mathsf{R}^*_\tau \cdot u \cdot w \cdot z \in \|\tau\|$. By a plain induction, it is possible to prove, for each numeral n, $\mathsf{R}^*_\tau \cdot u \cdot w \cdot n^* \in \|\tau\|$. Let s be a state and let $s' \geq s$ be such that $z \downarrow^{s'}$. Let $zs' = n$ with n numeral. Then for all $s'' \geq s'$,

$$(\mathsf{R}^*_\tau \cdot u \cdot w \cdot z)\, s'' = \mathsf{R}_\tau(us'')(vs'')(zs'') = \mathsf{R}_\tau(us'')(vs'')n = (\mathsf{R}^*_\tau \cdot u \cdot w \cdot n^*)s''$$

By Lemma 25, we conclude $\mathsf{R}^*_\tau \cdot u \cdot w \cdot z \in \|\tau\|$.

9. v is $\mathsf{min} : \mathtt{U} \to \mathtt{N}$. Suppose that $u \in \|U\|$. Let s be a state. Since $u \in \|U\|$, there exists $s' \geq s$ such that $u \downarrow^{s'}$. We have to prove that $\mathsf{min}^* \cdot u \in \|\mathtt{N}\|$. Let be $us' = \overline{U}$, with $\overline{U}$ update. For all $s'' \geq s'$:

$$(\mathsf{min}^* \cdot u)s'' = \mathsf{min}(us'') = \mathsf{min}\,\overline{U} = n$$

for some numeral n. By definition of $\|\mathtt{N}\|$, $\mathsf{min}^* \cdot u \in \|\mathtt{N}\|$.

10. v is $\Cup : \mathtt{U} \to \mathtt{U} \to \mathtt{U}$. Suppose that $u_1 \in \|\mathtt{U}\|$ and $u_2 \in \|\mathtt{U}\|$. We have to prove that $\Cup^* \cdot u_1 \cdot u_2 \in \|\mathtt{U}\|$. Let s be a state. Since $u_1 \in \|\mathtt{U}\|$ there exists $s' \geq s$ such that $u_1 \downarrow^{s'}$. Since $u_2 \in \|\mathtt{U}\|$, there exists $s'' \geq s'$ such that $u_2 \downarrow^{s''}$. Therefore, there exist two constants $\overline{U}_1$ and $\overline{U}_2$ such that for all $s''' \geq s''$, $u_1 s''' = \overline{U}_1$ and $u_2 s''' = \overline{U}_2$. Finally, for all $s''' \geq s''$,

$$(\Cup^* \cdot u_1 \cdot u_2)\, s''' = \Cup(u_1 s''')(u_2 s''') = \Cup\, \overline{U}_1 \overline{U}_2 = \overline{U}_3$$

for some update constant $\overline{U}_3$. By definition of $\|\mathtt{U}\|$, $\Cup^* \cdot u_1 \cdot u_2 \in \|\mathtt{U}\|$.

11. v is S, mkupd or get. The proof is similar to the one of the previous case.

12. v is a constant $\Phi_i : \mathtt{N} \to \mathtt{N}$ in $\mathcal{SC}$. Suppose now, for a term $u : \mathtt{N}$, that $u \in \|\mathtt{N}\|$. We have to prove that $\overline{\Phi}_i = (\lambda s^{\mathsf{S}}.s_i) \cdot u \in \|\mathtt{N}\|$. Let s be a state. We must show that there exists a $s' \geq s$ such that $(\lambda s^{\mathsf{S}}.s_i) \cdot u \downarrow^{s'}$. Since $u \in \|\mathtt{N}\|$, there exists a state $s' \geq s$ such that $u \downarrow^{s'}$. Let $n = us'$, with n numeral, and $m = s'_i(n)$. Let $\Phi_i = \Phi_{A(x,y)}$. If $A \notin \Gamma$, then trivially $(i, n) \in \mathsf{def}(s')$ by Definition 8. Therefore for all $s'' \geq s'$, $((\lambda s^{\mathsf{S}}.s_i) \cdot u)s'' = s''_i(n) = m$ and we are done. Hence, we may assume $A \in \Gamma$. There are two cases, and this is the only point of this proof in which we use EM.

a. $A(n, m)$ is true. Therefore, for all $s'' \geq s'$, $s''_i(n) = m$ because $(i, n) \in \mathsf{def}(s')$. Thus, for all $s'' \geq s'$, $((\lambda s^{\mathsf{S}}.s_i) \cdot u)s'' = s''_i(n) = m$, which is the thesis.

b. $A(n,m)$ is false. If there exists l such that $A(n,l)$ is true, then let

$$s'' := \lambda x^{\mathtt{N}} \lambda y^{\mathtt{N}}. \,\mathsf{if}\ x = i \wedge_{\mathtt{Bool}} y = n\ \mathsf{then}\ m\ \mathsf{else}\ s'_x(y)$$

Then, for all $s''' \geq s''$, $s'''_i(n) = l$ because $(i,l) \in \mathsf{def}(s'')$. Thus, for all $s''' \geq s''$,

$$((\lambda s^{\mathtt{S}}.s_i) \cdot u)s'' = s''_i(n) = m$$

which is the thesis. If there is no l such that $A(n,l)$ is true, then trivially $(i,n) \in \mathsf{def}(s')$. Thus for all $s'' \geq s'$, $((\lambda s^{\mathtt{S}}.s_i) \cdot u)s'' = s''_i(n) = m$ and we are done.

◀

The proofs of Proposition 17 and Theorem 18 remain exactly the same, while the proof of Theorem 19 can be straightforwardly adapted. In particular, in the base case of the induction one needs to prove that a term t, possibly with free variables of type $\mathtt{N}$, is computable. This follows from Theorem 26 and the fact that it is possible to prove by induction the statement $\forall x^{\mathtt{N}}.\ \lambda s^{\mathtt{S}} x \in \|\mathtt{N}\|$.

References

1 F. Aschieri, S. Berardi, *Interactive Learning-Based Realizability for Heyting Arithmetic with* EM_1, Logical Methods in Computer Science, 2010.
2 F. Aschieri, *Transfinite Update Procedures for Predicative Systems of Analysis*, Proc. of Computer Science Logic, 2011.
3 F. Aschieri, *A Constructive Analysis of Learning in Peano Arithmetic*, Annals of Pure and Applied Logic, 2011, doi:10.1016/j.apal.2011.12.004.
4 F. Aschieri, S. Berardi, *A New Use of Friedman's Translation: Interactive Realizability*, Festschrift of Helmut Schwichtenberg, Ontos-Verlag Series in Mathematical Logic, to appear.
5 F. Aschieri, *Learning Based Realizability for HA + EM1 and 1-Backtracking Games: Soundness and Completeness*, Annals of Pure and Applied Logic, to appear.
6 F. Aschieri, *Interactive Realizability for Second-Order Heyting Arithmetic with* EM_1 *and* SK_1, preprint, http://hal.inria.fr/hal-00657054.
7 F. Aschieri, *Interactive Realizability for Classical Peano Arithmetic with Skolem Axioms*, Technical Report, http://hal.inria.fr/hal-00685360.
8 J. Avigad, *Update Procedures and 1-Consistency of Arithmetic*, Mathematical Logic Quarterly, vol. 48, 2002.
9 J. Avigad, *Eliminating Definitions and Skolem Function in First-Order Logic*, ACM Transactions on Computational Logic, 2002.
10 S. Berardi and U. de' Liguoro, *Interactive Realizers. A New Approach to Program Extraction from non-Constructive Proofs*, ACM Transactions on Computational Logic, 2010.
11 T. Coquand, *A Semantic of Evidence for Classical Arithmetic*, Journal of Symbolic Logic, 1995.
12 H. Friedman, *Classically and Intuitionistically Provable Recursive Functions*, Lecture Notes in Mathematics, 1978, vol. 669/1978, 21-27.
13 K. Gödel, *Uber eine bisher noch nicht benutzte Erweiterung des finiten Standpunktes*, Dialectica 12, pp. 280-287, 1958.
14 J.-Y. Girard, *Proofs and Types*, Cambridge University Press,1989.
15 D. Hilbert, P. Bernays, *Grundalagen der Matematik*, vol II, Springer Berlin, 1939.
16 G. Kreisel, *On Weak Completeness of Intuitionistic Predicate Logic*, Journal of Symbolic Logic, vol. 27, 1962.

17 J.- L. Krivine, *Typed lambda-calculus in classical Zermelo-Fraenkel set theory*, Archive for Mathematical Logic, 40(3), 2001.

18 J.- L. Krivine, *Realizability Algebras 2: new models of ZF + DC*, Logical Methods in Computer Science, 2012.

19 G. Mints, S. Tupailo, W. Bucholz, *Epsilon Substitution Method for Elementary Analysis*, Archive for Mathematical Logic, vol. 35, 1996

20 M. H. Sorensen, P. Urzyczyn, *Lectures on the Curry-Howard isomorphism*, Studies in Logic and the Foundations of Mathematics, vol. 149, Elsevier, 2006.

21 W. Tait, *Intensional Interpretations of Functional of Finite Type*, The Journal of Symbolic Logic, 1967.

22 A. Troelstra, *Notions of Realizability for Intuitionistic Arithmetic and Intuitionistic Arithmetic in all Finite Types*, in Fenstad ed., Proc. of the Second Scandinavian Logic Symposium, North-Holland, 1972.

23 A. Troelstra, *Metamathematical Investigations of Intuitionistic Arithmetic and Analysis*, Lecture Notes in Mathematics, Springer-Verlag, Berlin-Heidelber-NewYork, 1973.

24 A. Troelstra, *Realizability*, in S. Buss ed., Handbook of Proof Theory, Studies in Logic and Foundations of Mathematics, Elsevier, 1998.

25 A. Troelstra, D. van Dalen, *Constructivism in Mathematics, vol. I*, North-Holland, 1988.