A Constructive Analysis of Learning in Peano Arithmetic

Federico Aschieri

Dipartimento di Informatica
Università di Torino
Italy

School of Electronic Engineering and Computer Science
Queen Mary, University of London
UK

Abstract

We give a constructive analysis of learning as it arises in various computational interpretations of classical Peano Arithmetic, such as Aschieri and Berardi learning based realizability, Avigad's update procedures and epsilon substitution method. In particular, we show how to compute in Gödel's system T upper bounds on the length of learning processes, which are themselves represented in T through learning based realizability. The result is achieved by the introduction of a new non standard model of Gödel's T, whose new basic objects are pairs of non standard natural numbers (convergent sequences of natural numbers) and moduli of convergence, where the latter are objects giving constructive information about the former. As foundational corollary, we obtain that that learning based realizability is a constructive interpretation of Heyting Arithmetic plus excluded middle over Σ_1^0 formulas (for which it was designed) and of all Peano Arithmetic when combined with Gödel's double negation translation. As byproduct of our approach, we also obtain a new proof of Avigad's theorem for update procedures and thus of termination of epsilon substitution method for PA.

Keywords: learning based realizability, classical arithmetic, update procedures, epsilon substitution method, constructive termination proof
2010 MSC: 03F03, 03F30, 03F55

1. Introduction

The aim of this paper is to carry out a detailed and complete constructive analysis of learning, as it arises in Aschieri and Berardi [2] learning based realizability for $\mathsf{HA}+\mathsf{EM}_1$ and in Avigad's [4] axiomatization of the epsilon substitution method for Peano Arithmetic through the concept of update procedure. The importance of this analysis is both practical and foundational. In the first place, we explicitly show how to compute upper bounds to the length of learning processes, thus providing the technology needed to analyze their computational complexity. Secondly, we answer positively to the foundational question of whether learning based realizability can be seen as an interpretation of classical Arithmetic into intuitionistic Arithmetic.

Learning based realizability is an extension of Kreisel modified realizability [15] to $\mathsf{HA} + \mathsf{EM}_1$ where

$$\mathsf{EM}_1 := \forall x^{\mathtt{N}}.\forall y^{\mathtt{N}}\neg Pxy \vee \exists y^{\mathtt{N}} Pxy$$

with Pxy decidable (the definition is given in section 6.3). In a few words, it is a way of making oracle computations more effective, through the use of approximations of oracle values and learning of new values by counterexamples. A learning based realizer is in the first place a term of Gödel's $\mathsf{T}_{\text{Class}}$, which is Gödel's system T plus an oracle $\Phi : \mathtt{N} \rightarrow \mathtt{N}$ of the same Turing degree of an oracle for the Halting problem. Of course, if a realizer was only this, it would be ineffective and so useless. Therefore, learning based realizers are computed with respect to *approximations* of the oracle Φ and thus effectiveness is recovered. Since approximations may be sometimes inadequate, results of computations may be wrong. But a learning based realizer is also a *self-correcting* program, able to spot incorrect oracle values used during computations and to correct them with *right* values. The new values are *learned*, surprisingly, by realizers of EM_1 and all the oracle values needed during each particular computation are acquired through learning processes. Here is the fundamental insight: classical principles may be computationally interpreted as learning devices.

An interesting link between learning based realizers and Avigad's update procedures is the following: realizers for atomic formulas are unary update procedures. We recall that a unary update procedure is a functional which takes as argument a function $f : \mathbb{N} \rightarrow \mathbb{N}$ - approximating some oracle - and it uses f to compute something. Then it tests whether the result of the computation is sound with respect to some target; if it is the case, it returns the empty set; otherwise, it returns a pair (n, m), meaning that f is ill-defined as an approximation of the oracle and must be updated as to output m on input n (and the pair (m, n) represents a correct association input-output for the oracle). The correspondence between realizers of atomic formulas and update procedures is important in many ways. First, using realizability, update procedures can now be extracted from classical proofs through Curry-Howard correspondence (see [19]) of proofs with terms of typed lambda calculi, without using Herbrand-type theorems or epsilon method as in Avigad [4]. This way, update procedures become suitable to be *actually* used in computational interpretations of actual proofs, without the need of examining hardly legible quantifier-free proofs. Moreover, thanks to the correspondence between update procedures and epsilon substitution method, the computational machinery of the latter is finally associated to a high level realizability semantics of proofs, which makes it much more understandable. Last, the powerful methods of type theory and realizability can be applied in order to reprove results on update procedures and epsilon substitution method in new ways.

Given the nature of learning based realizability as an extension of Kreisel modified realizability, our object of investigation is Gödel's system T and our metatheory will be purely intuitionistic. Our analysis will be accomplished by *restating* and then *reproving constructively* the following convergence theorem. First, given a term $s : \mathtt{N} \rightarrow (\mathtt{N} \rightarrow \mathtt{N})$ of Gödel's T and numerals $n \leq m$, define $s_n \preccurlyeq s_m$ iff for all numerals l, $s_n(l) \neq 0$ implies $s_n(l) = s_m(l)$ (see the premise to definition 3 for explanations). Then

Theorem 1 (Convergence). *Let $t : (\mathtt{N} \rightarrow \mathtt{N}) \rightarrow \mathtt{N}$ be a closed term of T. Let $s : \mathtt{N} \rightarrow (\mathtt{N} \rightarrow \mathtt{N})$ be any closed term of T representing a weakly increasing chain of functions:*

that is, assume that for every numerals $n \leq m$, $s_n \preccurlyeq s_m$ holds. Then, there exists an n such that for all $m \geq n$, $t(s_n) = t(s_m)$.

The intuitive meaning of the convergence theorem is the following. It is intended to be an analysis of oracle computations. That is, given a non computable function $f : \mathtt{N} \rightarrow \mathtt{N}$ one would like to "compute" $t(f)$. Since this is not effectively possible, in order to obtain significant results one may try nevertheless to define a weakly increasing chain s of functions with the property that for all numerals n, $s_n \preccurlyeq f$. Such a chain can be seen as a sequence of more and more refined approximations of f and can for example be constructed by means of *learning processes* as they arise in learning based realizability, Avigad's update procedures or epsilon substitution method (see Mints [16]). The theorem says that if t is computed with respect to such a sequence of approximations, then a stable answer about the value of $t(f)$ is eventually obtained.

The convergence theorem is already interesting in itself, but its special significance lies in its consequences, which we now describe and shall prove in the final part of the paper. Since most of them cannot be proved if the convergence theorem is not first restated and then proven constructively, they provide an important motivation for working in this direction.

A first consequence of the convergence theorem is that any learning process represented by a learning based realizer always terminates. We will express this by stating that any unary update procedure (see our definition in section 6.2, slightly different from Avigad's) representable in system T has a finite zero.

Theorem 2 (Zero Theorem for Unary Update Procedures, Informal). *Let $\mathcal{U} : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow ((\mathbb{N} \times \mathbb{N}) \cup \{\varnothing\})$ be a unary update procedure representable in system T. Then, it has a finite zero, i.e. there is a function f such that $\mathcal{U}(f) = \varnothing$.*

A zero of a unary update procedure represents a good approximation of the oracle that the procedure is trying to approximate. If the convergence theorem is proven constructively, also the above zero theorem can be. In particular, one can show that a zero of any update procedure $\mathcal{U}$ can be computed through a learning process guided by $\mathcal{U}$ itself. Moreover, one obtains also a constructive analysis of the number of learning steps required to complete the learning process through which the zero of $\mathcal{U}$ will be computed (see section 6.2). Let $\Vdash$ be the learning based realizability relation. The zero theorem constructively implies the following theorem:

Theorem 3 (Program Extraction via Learning Based Realizability). *Let t be a term of $\mathsf{T}_{\mathit{Class}}$ and suppose that $t \Vdash \forall x^{\mathtt{N}} \exists y^{\mathtt{N}} Pxy$, with Pxy atomic. Then, from t one can define a term u of Gödel's system T such that for every numeral n, $Pn(un) = \mathtt{True}$.*

The above theorem sharpens the result obtained in Aschieri and Berardi [2]. There, it has been proved as well that from any t such that $t \Vdash \forall x^{\mathtt{N}} \exists y^{\mathtt{N}} Pxy$ one can extract a computable function v such that for every numeral n, $Pn(vn) = \mathtt{True}$. However, the extracted v made use of unbounded iteration, while the u of theorem 3 is a "bounded" algorithm, that is, a program not explicitly using any kind of unbounded iteration. This is an important point from a foundational point of view: the algorithms extracted via learning based realizability *construct* witnesses, rather than *searching* for them.

As corollary, one obtains the important result that from classical proofs in Peano Arithmetic PA of $\forall\exists$-formulas one can extract bounded algorithms via learning based realizability $\Vdash$. This is done by first extracting a realizer from any given proof and then by applying theorem 3. In other words, one is able to give a novel proof of the following theorem due to Gödel (see [14]):

Theorem 4 (Provably Total Functions of PA). *If* $\mathsf{PA} \vdash \forall x^{\mathtt{N}} \exists y^{\mathtt{N}} Pxy$, *then there exists a term* u *of Gödel's system* T *such that for every numeral* n, $Pn(un) = \mathtt{True}$.

The novelty, here, is the technique employed to prove the theorem and the new understanding of extracted programs as realizers able to learn in a constructive way.

From a constructive proof of the convergence theorem one can also provide new constructive proofs of Avigad's [4] *fixed point theorem* for n-ary update procedures and hence of the *termination of the epsilon substitution method* for PA. Hence, one also obtains a *constructive analysis of learning* in Peano Arithmetic. The novelty, here, is the use of type theory to reason about the learning processes generated by update procedures and hence by epsilon substitution method. In particular, we solve a problem raised by Mints [17], asking for a Tait-style termination proof of the epsilon method for first order Peano Arithmetic.

Theorem 1 can be proven easily, but ineffectively, in second order logic:

Proof of theorem 1 (Ineffective). The informal idea of the proof is the following. Terms of system T use only a finite number of values of their function arguments. If we "apply" t to the least upper bound f_s of the sequence s (w.r.t the relation $\preccurlyeq$ of definition 3), we find that the finite part of f_s effectively used in the computation of $t(f_s)$ is already contained in some s_k. So, for every $h \geq k$, $t(s_h) = t(s_k)$.
Let use see the details. As proven by Kreisel (for a proof see Schwichtenberg [18]), t has a modulus of continuity $\mathcal{C}$, which is a term of system T of type $(\mathtt{N} \to \mathtt{N}) \to \mathtt{N}$ such that the following statement is provable in extensional HA^ω:

$$\forall f^{\mathtt{N}\to\mathtt{N}}, g^{\mathtt{N}\to\mathtt{N}}.(\forall x^{\mathtt{N}} \leq (\mathcal{C}f)\ f(x) = g(x)) \to t(f) = t(g) \tag{1}$$

By using the comprehension axiom, we can define the least upper bound f_s of the sequence s as follows

$$f_s(n) = \begin{cases} m & \text{if } \exists i \text{ such that } s_i(n) = m \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

Let $\mathcal{C}^M$ be the denotation of $\mathcal{C}$ in the full set theoretic model M of extensional HA^ω (see Kohlenbach [14]). Then there exists an n such that for all $m \geq n$

$$\forall x^{\mathtt{N}} \leq (\mathcal{C}^M f_s)\ s_n(x) = s_m(x)$$

By 1, we get that for all $m \geq n$, $t(s_n)^M = t(s_m)^M$. Hence by soundness of the model with respect to formal equality of extensional HA^ω, $t(s_n)$ and $t(s_m)$ normalize to the same numeral, since $t(s_n) = a$ and $t(s_m) = b$, with a, b numerals, implies $a^M = t(s_n)^M = t(s_m)^M = b^M$ and then $a = b$.

The convergence theorem is therefore true, but one cannot hope to prove it constructively as it is stated. In fact, it is a formula of the form $\forall\exists\forall$ and intuitionistic reasoning is already incomplete - when compared to classical reasoning - for that kind of formulas. It is known, for example, that classical finite type Peano Arithmetic PA^ω proves the formula $\forall f^{\mathtt{N}\to\mathtt{N}}\ \exists x^{\mathtt{N}}\ \forall y^{\mathtt{N}} f(x) \leq f(y)$, while intuitionistic Heyting Arithmetic HA^ω does not. In our case, one could associate to any Turing machine a weakly increasing sequence $s : \mathtt{N} \to (\mathtt{N} \to \mathtt{N})$ such that for all m, $s_m(n) = 0$ if $n \neq 0$, and $s_m(0) = 1$ if the machine terminates on input n in less than m steps, $s_m(0) = 0$ otherwise. A constructive proof of the convergence theorem relatively to the term $\lambda f^{\mathtt{N}\to\mathtt{N}} f(0)$ would compute the limit of the sequence $\lambda m^{\mathtt{N}} s_m(0)$, thus determining whether the Turing machine terminates on input n. By producing such a sequence s for every Turing machine, we would have a solution for the Halting problem.

Synopsis of the paper. In the rest of the paper, we develop a technology for constructively reasoning about convergence in Gödel's system T and proving a classically equivalent form of the convergence theorem. All proofs will be constructive and all their constructive content will be made explicit.

Our approach has a semantical content. In fact, we start from considering a kind of constructive non standard model for Peano Arithmetic and then we reinterpret Gödel's system T constants in order to manipulate the new individuals of the model. The reinterpretation of system T will turn out to be particularly suited to perform the computations we need to do for constructively reasoning about convergence. From the high level point of view, the proof techniques used amount to a combination of Kreisel's no-counterexample interpretation and Tait's reducibility/logical-relations method. With the first one, we can constructively reason about convergence. With the second, we prove the soundness of the model with respect to our purposes.

In detail, the plan of the paper is the following.

In section §2, we recall details of Gödel system T.

In section §3 we define the first ingredient of our approach, which is a constructive notion of convergence for sequences of objects, due to Berardi [6]. It is a no-counterexample interpretation of the classical notion of convergence, but it is different from the usual interpretation. Its main advantage is that it is very efficient from the computational point of view, since it enables *programming with continuations* and hence the writing of powerful and elegant realizers of its constructive content, which we will call *moduli of convergence*. Intuitively, a modulus of convergence for a convergent function $f : \mathtt{N} \to A$ will be a term able to find suitable intervals in which f is constant; moreover, the length of those intervals will depend on a continuation. At the end of the section we use Berardi's notion of convergence to reformulate the convergence theorem (see theorem 5).

In section §4, we introduce the second ingredient of our approach: a model that extends the usual set theoretic model of T generated over natural numbers by replacing naturals by pairs $\langle \mathcal{N}, f\rangle$ of a non standard number f (which is a function $\mathtt{N} \to \mathtt{N}$ as in ultrapower models of Peano Arithmetic) and its modulus of convergence $\mathcal{N}$. We also syntactically define a semantics $[\![\_]\!]_s$ (where s is a weakly increasing chain of functions) mapping terms of T in to elements of the model and in section §5 we show that, thanks to $[\![\_]\!]_s$, we can evaluate every term $t : (\mathtt{N} \to \mathtt{N}) \to \mathtt{N}$, into a pair $\langle \mathcal{N}, f\rangle$ such that $\mathcal{N}$ is a modulus of convergence for the function $f = \lambda n^{\mathtt{N}} t(s_n)$.

In section §6, we prove all the corollaries of the convergence theorem that we have discussed before.

2. Term Calculus

For a complete definition of Gödel's T we refer to Girard [11]. T is simply typed λ-calculus, with atomic types $\mathtt{N}$ (representing the set $\mathbb{N}$ of natural numbers) and $\mathtt{Bool}$ (representing the set $\mathbb{B} = \{\mathrm{True}, \mathrm{False}\}$ of booleans), product types $T \times U$ and arrows types $T \to U$, constants $0 : \mathtt{N}$, $\mathsf{S} : \mathtt{N} \to \mathtt{N}$, $\mathtt{True}, \mathtt{False} : \mathtt{Bool}$, pairs $\langle ., . \rangle$, projections π_0, π_1, conditional if_T and primitive recursion R_T in all types, and the usual reduction rules $(\beta), (\pi), (\mathsf{if}), (\mathsf{R})$ for λ, $\langle ., . \rangle, \mathsf{if}_T, \mathsf{R}_T$. From now on, if t, u are terms of T with $t = u$ we denote provable intensional equality in T. If $k \in \mathbb{N}$, the numeral denoting k is the closed normal term $\mathsf{S}^k(0)$ of type $\mathtt{N}$. Terms of the form $\mathsf{if}_T t_1 t_2 t_3$ will be written in the more legible form $\mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3$. All closed normal terms of type $\mathtt{N}$ are numerals. Any closed normal term of type $\mathtt{Bool}$ in $\mathcal{T}$ is $\mathtt{True}$ or $\mathtt{False}$.

Notation. For notational convenience and to define in a more readable way terms of type $A \times B \to C$, for any variables $x_0 : A$ and $x_1 : B$ we define

$$\lambda \langle x_0, x_1 \rangle^{A \times B} u := \lambda x^{A \times B} u[\pi_0 x / x_0\ \pi_1 x / x_1]$$

where x is a fresh variable not appearing in u. We observe that for any terms t_0, t_1

$$(\lambda \langle x_0, x_1 \rangle^{A \times B} u) \langle t_0, t_1 \rangle = u[t_0 / x_0\ t_1 / x_0]$$

Often, it is useful to add to system T new constants and atomic types, together with a set of algebraic reduction rules we call "functional".

Definition 1 (Functional set of rules). *Let C be any set of constants, each one of some type $A_1 \to \ldots \to A_n \to A$, for some atomic types $A_1, \ldots, A_n, A$. We say that $\mathcal{R}$ is a* functional set of reduction rules *for C if $\mathcal{R}$ consists, for all $c \in C$ and all closed normal terms $a_1 : A_1, \ldots, a_n : A_n$ of $\mathcal{T}$, of one and exactly one rule $c a_1 \ldots a_n \mapsto a$, where $a : A$ is a closed normal term of $\mathcal{T}$.*

If a system $\mathcal{T}$ is obtained from Gödel's T by adding a recursive set C of constants and a recursive functional set of rules for C, we we call $\mathcal{T}$ a *simple extension of* T. $\mathcal{T}$ is strongly normalizing, by standard reducibility arguments (see e.g. Berger [10]). Any atomic-type term of any simple extension $\mathcal{T}$ of T is equal either to a numeral, if it is of type $\mathtt{N}$, or to a boolean, if it is of type $\mathtt{Bool}$, or to a constant of type A, if it is of type A. All results of this paper hold whatever simple extension of T is chosen. Let us fix one.

Definition 2 (System $\mathcal{T}$). *From now on, we denote with $\mathcal{T}$ be an arbitrarily chosen simple extension of Gödel's system T. We also assume that $\mathcal{T}$ contains constants for deciding equality of constants of atomic type.*

Throughout the paper, the intended interpretation of the natural number 0 will be as a "default" value. That is, when we do not have any information about what value a function has on argument n, we assume that it has value 0. That being said, it is natural to consider a function $f_2 : \mathbb{N} \to \mathbb{N}$ to be *extending* another function $f_1 : \mathbb{N} \to \mathbb{N}$, whenever it holds that for every n such that $f_1(n)$ is a non default value (and hence different from 0), then $f_1(n) = f_2(n)$. f_2 may hence have a non default value at some argument f_1 has a default value, but it agrees with f_1 at the arguments f_1 has not a default value. So, f_2 carries more information than f_1.

Definition 3 (Ordering Between Functions and Terms). *Let f_1, f_2 be functions $\mathbb{N} \to \mathbb{N}$. We define*

$$f_1 \preccurlyeq f_2 \iff \forall n \in \mathbb{N}\ f_1(n) \neq 0 \Rightarrow f_1(n) = f_2(n)$$

Moreover, if t_1, t_2 are closed terms of $\mathcal{T}$ of type $\mathtt{N} \to \mathtt{N}$ representing respectively functions $g_1, g_2 : \mathbb{N} \to \mathbb{N}$, we will write $t_1 \preccurlyeq t_2$ if and only if $g_1 \preccurlyeq g_2$.

In the following, we will write "$s \in$ w.i." if $s : \mathtt{N} \to (\mathtt{N} \to \mathtt{N})$ is a closed term representing a weakly increasing sequence of functions, that is, if for all numerals n, m, $n \leq m$ implies $s_n \preccurlyeq s_m$.

3. The No-Counterexample Interpretation and Berardi's Notion of Convergence

In this paper, we are interested in arithmetical formulas stating convergence of natural number sequences. Classically, we consider a sequence of natural numbers to be convergent if it is definitely constant, that is, if the there is an element of the sequence which is equal to all successive elements of the sequence. Hence, we will consider formulas of the form

$$(\forall z^A)\ \exists x^{\mathtt{N}} \forall y^{\mathtt{N}} P(z, x, y) \tag{2}$$

Since that kind of formulas cannot generally be proven constructively, a common standpoint is to consider classically equivalent but constructively weak enough statements, as in Kreisel *no-counterexample interpretation*:

$$(\forall z^A)\ \forall f^{\mathtt{N} \to \mathtt{N}} \exists x^{\mathtt{N}} P(z, x, f(x))$$

If the statement (2) (with $A = \mathtt{N}$) is provable in PA, then one can constructively extract from any proof a term $t : (\mathtt{N} \to \mathtt{N}) \to \mathtt{N}$ of Gödel's system T such that

$$(\forall z^{\mathtt{N}})\ \forall f^{\mathtt{N} \to \mathtt{N}} P(z, t(f), f(t(f)))$$

holds (see for example Kohlenbach [14]). In our cases, we have to deal with formulas of the form

$$\exists x^{\mathtt{N}}\ \forall y^{\mathtt{N}} \geq x f(x) = f(y)$$

where f is a term of type $\mathtt{N} \to \mathtt{N}$, and hence we may be tempted to consider their no-counterexample interpretation

$$\forall h^{\mathtt{N} \to \mathtt{N}}\ \exists x^{\mathtt{N}}\ h(x) \geq x \to f(x) = f(h(x)) \tag{3}$$

If one introduces the notation

$$f \downarrow [n, m] \overset{\text{def}}{\equiv} \forall x^{\mathtt{N}}.\ n \leq x \leq m \to f(x) = f(n)$$

one often finds in literature the following equivalent version of (3):

$$\forall h^{\mathtt{N} \to \mathtt{N}} \exists x^{\mathtt{N}}\ f \downarrow [x, h(x)] \tag{4}$$

which is the no-counterexample interpretation of

$$\exists x^{\mathtt{N}}\ \forall y^{\mathtt{N}} \geq x\ f \downarrow [x, y]$$

While the above notion of convergence (4) would be enough for our purposes, it seems not to allow straightforward compositional reasoning when one has to deal with non trivial *interaction* of convergent functions. Even when there is no complex interaction, the needed reasoning is not direct. For example, one may want to prove that if two functions f, g converge in the sense of (4), one can systematically find intervals in which they are *both* constant. That is, if

$$\forall h^{\mathtt{N}\to\mathtt{N}}\ \exists x^{\mathtt{N}}\ f \downarrow [x, h(x)] \wedge \forall h^{\mathtt{N}\to\mathtt{N}}\ \exists x^{\mathtt{N}}\ g \downarrow [x, h(x)]$$

then one may want to prove that

$$\forall h^{\mathtt{N}\to\mathtt{N}}\ \exists x^{\mathtt{N}}\ f \downarrow [x, h(x)] \wedge g \downarrow [x, h(x)]$$

The above implication is provable in a non overly complicated way, but when interaction increases (as we shall see in proposition 2 below), one begins to feel the need for a more suitable formulation of convergence.

Berardi[1] [6] introduced a notion of convergence especially suited for managing interaction of convergent functions. If one consider the formula

$$\forall z^{\mathtt{N}}\ \exists x^{\mathtt{N}} \geq z\ \forall y^{\mathtt{N}} \geq x f \downarrow [x, y]$$

(with the intent of expressing very redundantly the fact that there are infinite points of convergence for f) one obtains a very strong notion of constructive convergence by taking its no-counterexample interpretation

$$\forall z^{\mathtt{N}}\ \forall h^{\mathtt{N}\to\mathtt{N}}\ \exists x^{\mathtt{N}}\ \geq z\ h(x) \geq x \to f \downarrow [x, h(x)]$$

which after skolemization becomes

$$\forall h^{\mathtt{N}\to\mathtt{N}}\ \exists \alpha^{\mathtt{N}\to\mathtt{N}} \geq \mathsf{id}\ \forall z^{\mathtt{N}}\ h(z) \geq z \to f \downarrow [\alpha(z), h(\alpha(z))]$$

which is equivalent to

$$\forall h^{\mathtt{N}\to\mathtt{N}} \geq \mathsf{id}\ \exists \alpha^{\mathtt{N}\to\mathtt{N}} \geq \mathsf{id}\ \forall z^{\mathtt{N}} f \downarrow [\alpha(z), h(\alpha(z))] \tag{5}$$

where we have used the notation

$$\alpha^{\mathtt{N}\to\mathtt{N}} \geq \mathsf{id} \overset{\text{def}}{\equiv} \forall x^{\mathtt{N}} \alpha(x) \geq x$$

We observe that (4) and (5) are constructively equivalent. However, from a computational point of view, their realizers are quite different: the realizers of (5) are able to interact directly with each other, as we will see.

We are now ready to formally define a constructive notion of convergence for sequences of numbers: a sequence of objects $f : \mathtt{N} \to A$ is convergent if for any $h^{\mathtt{N}\to\mathtt{N}} \geq \mathsf{id}$ there are infinitely many intervals $[n, h(n)]$ in which f is constant.

[1] We thank Stefano Berardi for letting us to apply his unpublished notion of convergence in this work

Definition 4 (Convergence (Berardi [6])). *Let $f : \mathtt{N} \to A$ be a closed term of $\mathcal{T}$, with A atomic type. We say that f* converges *if*

$$\forall h^{\mathtt{N}\to\mathtt{N}} \geq \mathsf{id}\ \exists \alpha^{\mathtt{N}\to\mathtt{N}} \geq \mathsf{id}\ \forall z^{\mathtt{N}} f \downarrow [\alpha(z), h(\alpha(z))]$$

Notation. If $t : A \to B$ and $u : A$ we shall often write t_u in place of tu, for notational convenience or for highlighting that t_u is an element of a collection of type-B terms parametrized by terms of type A.

We now make explicit the constructive information associated to the above notion of convergence, through the concept of *modulus of convergence*[2]. A modulus of convergence takes an $h : \mathtt{N} \to \mathtt{N}$ and returns an enumeration of intervals $[n, h(n)]$ in which f is constant. It is a intuitionistic realizer of the notion of convergence.

Definition 5 (Modulus of Convergence). *Let $f : \mathtt{N} \to A$ be a closed term of $\mathcal{T}$, with A atomic type. A term $\mathcal{M} : (\mathtt{N} \to \mathtt{N}) \to (\mathtt{N} \to \mathtt{N})$ of $\mathcal{T}$ is a* modulus of convergence *for f if*

1. $\forall h^{\mathtt{N}\to\mathtt{N}} \geq \mathsf{id}\ \mathcal{M}_h \geq \mathsf{id}$
2. $\forall h^{\mathtt{N}\to\mathtt{N}} \geq \mathsf{id}\ \forall z^{\mathtt{N}} f \downarrow [\mathcal{M}_h(z), h(\mathcal{M}_h(z))]$

*If $h : \mathtt{N} \to \mathtt{N} \geq \mathsf{id}$ and $\forall z^{\mathtt{N}} f \downarrow [\mathcal{N}(z), h(\mathcal{N}(z))]$, $\mathcal{N}$ is said to be an h-*modulus of convergence *for f.*

We observe that by definition, if one has a modulus of convergence $\mathcal{M}$ for a function f, he can find an infinite number of intervals of any desired length in which f is constant. For example, if one wants to find an interval of length 5, he just define the function $h(x) = x + 5$ and compute $n := \mathcal{M}_h(0)$. Then, f is constant in $[n, n + 5]$. Clearly, a modulus of convergence carries a lot of constructive information about f.

3.1. Intuitive Significance of the Concept of Modulus of Convergence and Restatement of the Convergence Theorem

As we said, Berardi's notion of convergence works remarkably well when convergent functions interact together, for instance, in the definition of a new function. The fact that Berardi's notion is a no-counterexample interpretation of the classical notion of convergence, explains why it *works.* We can intuitively describe the reasons why it does it *well* as follows.

A first reason is purely computational. Given a function $h^{\mathtt{N}\to\mathtt{N}} \geq \mathsf{id}$ and a modulus of convergence $\mathcal{M}$, we can interpret the role of h in the computation of $\mathcal{M}_h$ as that of a *continuation.* Constructively, when a new convergent function is defined from other convergent functions, one will need to produce intervals in which the new function is constant. Thus, he may try to achieve the goal by finding intervals in which the functions

[2] Which is not to be confounded with the notion of modulus of convergence as used in real analysis: ours is a constructive variant. Since however the notion of convergence we are realizing is classically equivalent to the usual one, we keep the terminology.

involved in the definition are *all* constant. The problem is that he may be able to find such intervals for every single function, but not for them all together. For example, if we define the function

$$\beta := \lambda x^{\mathtt{N}} f(g(x), x)$$

then β is convergent if g and $\lambda x^{\mathtt{N}} f(n, x)$ are such for every choice of n. But an interval in which g is constant need not be an interval in which β too is constant, because we have to find some interval in which *both* g is constantly equal to some m and $\lambda x^{\mathtt{N}} f(m, x)$ is constant. We solve the problem through the use of continuations.

We start by observing that it seems there is a strict sequence of tasks to be performed. First, one tries to find an m_1 such that g is constant in, say, $[m_1, l_1]$ with $m_1 < l_1$. Then, he computes $g(m_1) = n$ and pass n to a "continuation" $h : \mathtt{N} \to \mathtt{N}$ which returns an $h(n) = m_2 < l_2$ such that $\lambda x^{\mathtt{N}} f(n, x)$ is constant in $[m_2, l_2]$. If $m_1 < m_2 < l_1$, a non trivial interval in which g is constant has been found. But if $m_2 > l_1$? Then, g may assume different values in all points of the interval $[m_2, l_2]$ and one cannot hope that β is going to be convergent in $[m_2, l_2]$. We anticipate the solution contained in the proof of proposition 2, by letting $m_1 = \mathcal{M}_k(0)$, where $\mathcal{M}$ is a modulus of convergence for g and, for example,

$$h'(x) = h(g(x)) + 1$$

Then, by definition of modulus of convergence, g is constant in $[m_1, h'(m_1)]$ and letting $l_1 = h'(m_1)$ we obtain that

$$m_2 = h(n) = h(g(m_1)) < h(g(m_1)) + 1 = l_1$$

as required. In other words, we use h' and hence h as *continuations*, thanks to $\mathcal{M}$.

The issue we are facing may be further exemplified by the following sequential game between k players. Suppose there are convergent functions $f_1, f_2, \ldots, f_k$ ot type $\mathtt{N} \to \mathtt{N}$ on the board and an arbitrarily chosen number m. Players make their moves in order, starting from player one and finishing with player k. A play of the game is an increasing sequence of numbers $m, m_1, m_2, \ldots, m_k$, with m_i the move of player i. Player i wins if f_i is constant in an interval $[m_k, l_k]$, for some $l_k > m_k$. A strategy for player i is just a function h over natural numbers, taking the move of the player $i - 1$ (or the integer m if $i = 1$) and returning the move of player i. The fact that the winning condition depends on the move of player k makes very difficult for players $1, \ldots, k-1$ to win. In this game, each player hopes that in the resulting final interval its own function will be constant but his hope is frustrated by the following ones, which are trying to accomplish the same task but with respect to their own functions. However, as we will show, player i in some cases may have a winning strategy effectively computable if he knows the strategies of all subsequent players $i + 1, \ldots, k$, *even* if he does not know how to compute a point of stability of his own function.

We are now in a position to tell another reason why moduli of convergence are so useful. A winning strategy for player i can be computed by a convergence modulus. More precisely, it can be proved, as consequence of proposition 1, that if players $i+1, \ldots, k$ play strategies $h_{i+1}, \ldots, h_k$, then $h_i := \mathcal{M}_{h_k \circ \cdots \circ h_{i+1}}$ is a winning strategy for player i against $h_{i+1}, \ldots, h_k$, whenever $\mathcal{M}$ is a modulus of convergence for f_i. Therefore, if a modulus of convergence for each function $f_1, \ldots, f_k$ is given, one can compute a particularly desirable instance of *Nash equilibrium*, that is, a sequence of functions $h_1, h_2, \ldots, h_k$ such that, if every player i plays according to the strategy h_i, every play will be won by every player. Therefore, at the end of the interaction, every participant will have accomplished its own task.

We now formulate the promised restatement of theorem 1 that we shall be able to prove.

Theorem 5 (Weak Convergence). *Let $t : (\mathtt{N} \to \mathtt{N}) \to \mathtt{S}$ be a closed term of $\mathcal{T}$, with $\mathtt{S}$ atomic type. Then we can effectively define a closed term $\mathcal{M} : (\mathtt{N} \to (\mathtt{N} \to \mathtt{N})) \to (\mathtt{N} \to \mathtt{N}) \to (\mathtt{N} \to \mathtt{N})$ of $\mathcal{T}$, such that the following holds: for all $s : \mathtt{N} \to (\mathtt{N} \to \mathtt{N})$ such that $s \in$ w.i., $\mathcal{M}s$ is a modulus of convergence for $\lambda m^{\mathtt{N}} t(s_m)$.*

3.2. Basic Operations with Moduli of Convergence

We now prove a couple of propositions, both to illustrate the use of moduli of convergence and to provide lemmas we will need in the following. First, we show that given two terms f_1 and f_2, if each one of them has a modulus of convergence, then there is a modulus of convergence that works simultaneously for both of them. In particular, we can define a binary operation $\sqcup$ between moduli of convergence such that, for every pair of moduli $\mathcal{M}, \mathcal{N}$, $\mathcal{M} \sqcup \mathcal{N}$ is "more general" than both $\mathcal{M}$ and $\mathcal{N}$. Here, for every $\mathcal{M}_1, \mathcal{M}_2$, we call $\mathcal{M}_2$ more general than $\mathcal{M}_1$, if for every term f, if $\mathcal{M}_1$ is a modulus of convergence for f then also $\mathcal{M}_2$ is a modulus of convergence for f. We this terminology, we may see $\mathcal{M} \sqcup \mathcal{N}$ as an upper bound of the set $\{\mathcal{M}, \mathcal{N}\}$, with respect to the partial order induced by the relation "to be more general than". The construction of the pair $\mathcal{M}_{h \circ \mathcal{N}_h}, \mathcal{N}_h$ below may also be seen as a Nash equilibrium for the two player version of the game we have discussed above.

Proposition 1 (Joint Convergence). *Let $\mathcal{M}$ and $\mathcal{N}$ be moduli of convergence respectively for f_1 and f_2. Define*

$$\mathcal{M} \sqcup \mathcal{N} := \lambda h^{\mathtt{N} \to \mathtt{N}} \lambda z^{\mathtt{N}} \mathcal{N}_h(\mathcal{M}_{h \circ \mathcal{N}_h}(z))$$

Then $\mathcal{M} \sqcup \mathcal{N}$ is a modulus of convergence for both f_1 and f_2.

Proof. Set

$$\mathcal{L} := \mathcal{M} \sqcup \mathcal{N}$$

First, we check property 1 of definition 5. For all $h^{\mathtt{N} \to \mathtt{N}} \geq \mathsf{id}$, $\mathcal{N}_h \geq \mathsf{id}$ by definition 5 point (1) and so $h \circ \mathcal{N}_h \geq \mathsf{id}$. Thus, for all $h^{\mathtt{N} \to \mathtt{N}} \geq \mathsf{id}$ and $z^{\mathtt{N}}$

$$\mathcal{L}_h(z) = \mathcal{N}_h(\mathcal{M}_{h \circ \mathcal{N}_h}(z)) \geq z$$

since $\mathcal{M}$ has property (1) of definition 5 and hence $\mathcal{M}_{h \circ \mathcal{N}_h} \geq \mathsf{id}$. Therefore, for all $h^{\mathtt{N} \to \mathtt{N}} \geq \mathsf{id}$, $\mathcal{L}_h \geq \mathsf{id}$ and we are done.
Secondly, we check property 2 of definition 5. Fix a term $h^{\mathtt{N} \to \mathtt{N}} \geq \mathsf{id}$ and a numeral z. We have that

$$f_1 \downarrow [\mathcal{M}_{h \circ \mathcal{N}_h}(z), h \circ \mathcal{N}_h(\mathcal{M}_{h \circ \mathcal{N}_h}(z))] \tag{6}$$

since $\mathcal{M}$ is a module of convergence for f_1. Moreover,

$$f_2 \downarrow [\mathcal{N}_h(\mathcal{M}_{h \circ \mathcal{N}_h}(z)), h(\mathcal{N}_h(\mathcal{M}_{h \circ \mathcal{N}_h}(z)))] \tag{7}$$

since $\mathcal{N}$ is a modulus of convergence for f_2. But the starting point of the interval in (7) is greater or equal to the starting point of the interval in (6), for $\mathcal{N}_h \geq \mathsf{id}$, while their ending points are equal. Hence also

$$f_1 \downarrow [\mathcal{N}_h(\mathcal{M}_{h\circ\mathcal{N}_h}(z)), h(\mathcal{N}_h(\mathcal{M}_{h\circ\mathcal{N}_h}(z)))]$$

and hence both f_1 and f_2 are constant in the interval $[\mathcal{L}_h(z), h(\mathcal{L}_h(z))]$ by definition of $\mathcal{L}$.

We now consider a situation in which a family $\{f_n\}_{n\in\mathtt{N}}$ of convergent terms interacts with a convergent term g and we show the result of the interaction is still a convergent term. In the following, we call "object of type A" any closed normal term of type A.

Proposition 2 (Merging of Functions). *Let $f : A \to (\mathtt{N} \to A)$ be a closed term, with A atomic, and $\mathcal{N} : A \to (\mathtt{N} \to \mathtt{N}) \to (\mathtt{N} \to \mathtt{N})$ be such that for every object a of type A, $\mathcal{N}_a$ is a modulus of convergence for f_a. Let moreover $g : \mathtt{N} \to A$ and let $\mathcal{M}$ be a modulus of convergence for g. Define*

$$\mathcal{H}_1(\mathcal{M}, \mathcal{N}, g) := \lambda h^{\mathtt{N}\to\mathtt{N}} \lambda z^{\mathtt{N}} \mathcal{N}'_h(\mathcal{M}_{h\circ\mathcal{N}'_h}(z))$$

with

$$\mathcal{N}'_h := \lambda n^{\mathtt{N}} (\mathcal{N}_{g(n)} h) n$$

Then $\mathcal{H}_1(\mathcal{M}, \mathcal{N}, g)$ is a modulus of convergence for

$$\lambda n^{\mathtt{N}} f_{g(n)}(n)$$

Proof. Property (1) of definition 5 follows by the same reasoning used in proposition 1. We check property (2) of definition 5. Set $\mathcal{L} := \mathcal{H}_1(\mathcal{M}, \mathcal{N}, g)$. The idea is that $\mathcal{L}$ has to produce an interval i in which g is constant and equal to a, while the interval produced by $\mathcal{N}_a$ in which f_a is constant will be contained in i. $\mathcal{L}$ does the job by using $\mathcal{N}'_h$ as a *continuation.*
Fix a term closed $h^{\mathtt{N}\to\mathtt{N}} \geq \mathsf{id}$ and z a numeral. We have that

$$g \downarrow [\mathcal{M}_{h\circ\mathcal{N}'_h}(z), h \circ \mathcal{N}'_h(\mathcal{M}_{h\circ\mathcal{N}'_h}(z))] \tag{8}$$

since $\mathcal{M}$ is a module of convergence for g. In particular,

$$g \downarrow [\mathcal{N}'_h(\mathcal{M}_{h\circ\mathcal{N}'_h}(z)), h(\mathcal{N}'_h(\mathcal{M}_{h\circ\mathcal{N}'_h}(z)))] \tag{9}$$

since $\mathcal{N}'_h \geq \mathsf{id}$. Say that for all n in the intervals in (8) and (9), $g(n) = a$. By definition of $\mathcal{N}'_h$

$$[\mathcal{N}'_h(\mathcal{M}_{h\circ\mathcal{N}'_h}(z)), h(\mathcal{N}'_h(\mathcal{M}_{h\circ\mathcal{N}'_h}(z)))]$$

$$= [\mathcal{N}_a h(\mathcal{M}_{h\circ\mathcal{N}'_h}(z)), h(\mathcal{N}_a h(\mathcal{M}_{h\circ\mathcal{N}'_h}(z)))] \tag{10}$$

Since $\mathcal{N}_a$ is a modulus of convergence for f_a, we have

$$f_a \downarrow [\mathcal{N}_a h(\mathcal{M}_{h\circ\mathcal{N}'_h}(z)), h(\mathcal{N}_a h(\mathcal{M}_{h\circ\mathcal{N}'_h}(z)))]$$

But for all x in the interval (10),

$$(\lambda n^{\mathtt{N}} f_{g(n)}(n))x = f_a(x)$$

Hence

$$\lambda n^{\mathtt{N}} f_{g(n)}(n) \downarrow [\mathcal{N}'_h(\mathcal{M}_{h\circ\mathcal{N}'_h}(z)), h(\mathcal{N}'_h(\mathcal{M}_{h\circ\mathcal{N}'_h}(z)))]$$

and so $\lambda n^{\mathtt{N}} f_{g(n)}(n)$ is constant in the interval $[\mathcal{L}_h(z), h(\mathcal{L}_h(z))]$ by definition of $\mathcal{L}$.

4. Computations with non Standard Natural Numbers

For technical convenience we add now to system $\mathcal{T}$ a constant $\Phi : \mathtt{N} \to \mathtt{N}$ with no associated reduction rules. In this way, each term $t : A$ can be viewed as functionally depending on Φ, but it is still considered as having type A, instead the more complicated $(\mathtt{N} \to \mathtt{N}) \to A$. Of course, terms of atomic type are not in general equal to a constant or a numeral, if they contain Φ.

Definition 6 (Evaluation at u). *Let t be a term. For any term $u : \mathtt{N} \to \mathtt{N}$, we denote with $t[u]$ the term $t[u/\Phi]$.*

Adopting this notation, what we want prove is that if $t : A$, with A atomic, and $s \in$ w.i., then the function $\lambda m^{\mathtt{N}} t[s_m]$ constructively converges, that is, it has a modulus of convergence. A natural attempt for achieving the goal is to recursively decompose the problem. For example, suppose we want to study the convergence of the function

$$^*(+t_1t_2) := \lambda m^{\mathtt{N}} + t_1t_2[s_m] : \mathtt{N} \to \mathtt{N}$$

where $s \in$ w.i.. and $+ : \mathtt{N} \to \mathtt{N} \to \mathtt{N}$ represents a constant of $\mathcal{T}$ encoding the operation of addition of natural numbers. Since $t_1 : \mathtt{N}$ and $t_2 : \mathtt{N}$ may have complex structure, it is natural to recursively study the functions

$$^*t_1 := \lambda m^{\mathtt{N}} t_1[s_m] : \mathtt{N} \to \mathtt{N}$$

and

$$^*t_2 := \lambda m^{\mathtt{N}} t_2[s_m] : \mathtt{N} \to \mathtt{N}$$

But if we want to study the function $^*(+t_1t_2)$ as a combination of $^*t_1$ and $^*t_2$, it is clear that $+$ cannot be interpreted as itself, but as a function $^*+$ of $^*t_1$ and $^*t_2$. We would like the following equation to hold

$$^*(+t_1t_2) = {^*+}{^*t_1^*}t_2$$

As a consequence of our notation, also the following equation must be true for all numerals n

$$^*(+t_1t_2)(n) = +(^*t_1(n))(^*t_2(n))$$

These considerations impose us to define

$$^*+ := \lambda g_1^{\mathtt{N}\to\mathtt{N}} \lambda g_2^{\mathtt{N}\to\mathtt{N}} \lambda m^{\mathtt{N}} + g_1(n)g_2(n)$$

At a first look, this may seem a rather strange way of doing computations. But it turns out that it is *strongly* not the case. $^*t_1$ and $^*t_2$ may be interpreted as *hypernatural* numbers and $^*+$ as the operation of addition of hypernaturals as they are defined in ultrapower non standard models of Peano Arithmetic.

4.1. Non Standard Models of Arithmetic

The first non standard model of Arithmetic is due to Skolem [20]. The universe of that model is indeed made of functions $\mathbb{N} \to \mathbb{N}$, but we instead describe a variant of the Skolem construction, which is the ultrapower construction (see for example Goldblatt [12]).

Fix a non principal ultrafilter $\mathcal{F}$ over $\mathbb{N}$. First, define an equivalence relation $\simeq$ between functions $\mathbb{N} \to \mathbb{N}$ as follows:

$$f_1 \simeq f_2 \iff \{x \in \mathbb{N} \mid f_1(x) = f_2(x)\} \in \mathcal{F}$$

(The intuition here is that an ultrafilter collects the "big" subsets of $\mathbb{N}$ and hence two functions are to be considered equal if they have equal values for "great many" arguments. For example, two functions which, as sequences, converge to the same natural number are considered equal, for they agree on a cofinite set of $\mathbb{N}$, which must belong to every non principal ultrafilter). Secondly, define

$$^*\mathbb{N} := (\mathbb{N} \to \mathbb{N})_{\simeq}$$

that is, $^*\mathbb{N}$ is the set of all natural number functions partitioned under the equivalence relation $\simeq$. Finally, set

$$\begin{aligned}
^*0 &:= \lambda n^{\mathbb{N}} 0 \\
^*S &:= \lambda n^{\mathbb{N}} S(n) \\
^*+ &:= \lambda f_1^{\mathbb{N}\to\mathbb{N}} \lambda f_2^{\mathbb{N}\to\mathbb{N}} \lambda n^{\mathbb{N}} f_1(n) + f_2(n) \\
^*\cdot &:= \lambda f_1^{\mathbb{N}\to\mathbb{N}} \lambda f_2^{\mathbb{N}\to\mathbb{N}} \lambda n^{\mathbb{N}} f_1(n) \cdot f_2(n)
\end{aligned}$$

where $S, +, \cdot$ are the usual operations over natural numbers. In general, if one wants to define the non standard version of a standard function $f : \mathbb{N}^k \to \mathbb{N}$, he simply lets

$$^*f := \lambda f_1^{\mathbb{N}\to\mathbb{N}} \dots \lambda f_k^{\mathbb{N}\to\mathbb{N}} \lambda n^{\mathbb{N}} f(f_1(n), \dots, f_k(n))$$

It can be proved that the structure

$$(^*\mathbb{N}, {}^*0, {}^*S, {}^*+, {}^*\cdot)$$

is a model of Peano Arithmetic as similar to the usual structure of natural numbers as to satisfy *precisely* the same sentences which are true under the usual interpretation. Formally, it is *elementarily equivalent* to the structure of natural numbers.

Elements of $^*\mathbb{N}$ are usually called hypernatural numbers. Since they are so similar to natural numbers, it perfectly makes sense to think about defining a model of system $\mathcal{T}$ over hypernaturals. Indeed, Berardi [7] used hypernaturals, under a weaker equivalence relation, to construct an intuitionistic model for Δ^2_0 maps and Berardi and de' Liguoro [5] used them to interpret a fragment of classical primitive recursive Arithmetic.

4.2. A non Standard Model for the System T_0

In order to approach gradually our final construction, we first give a definition of a non standard model for T_0, which is Gödel's T restricted to having only a recursion operator

R of type $\mathtt{N} \to (\mathtt{N} \to \mathtt{N} \to \mathtt{N}) \to \mathtt{N} \to \mathtt{N}$ and choice operator $\mathsf{if} : \mathtt{Bool} \to \mathtt{N} \to \mathtt{N} \to \mathtt{N}$. Hence, T_0 represents the primitive recursive functions.

The definition of the model and of the interpretation function $\_^*$ is purely syntactical and this is the key for our approach to go through. In other words, the interpretation of a type will be a type and the interpretation of a term will remain a term. It will be evident that from our syntactical definitions one obtains also a set-theoretic semantics. But in fact, we are defining an internal model, that is a representation of T_0 into T itself. First, define the new type structure as:

$$\begin{aligned} {}^*\mathtt{N} &:= \mathtt{N} \to \mathtt{N} \\ {}^*\mathtt{Bool} &:= \mathtt{N} \to \mathtt{Bool} \\ {}^*(A \to B) &:= {}^*A \to {}^*B \\ {}^*(A \times B) &:= {}^*A \times {}^*B \end{aligned}$$

From a semantical point of view, we interpret natural numbers as functions. Since the construction is syntactical, there is no need to describe an equivalence relation between those functions. But, accordingly to which equivalence relation one has in mind, the definition we are going to give will make sense or not from the *semantical* point of view. For the results of this paper, we have no utility in putting extra effort to define a model for T_0, which is also a model for Peano Arithmetic. Hence, for now we may assume that ${}^*\mathtt{N}$ represents just all functions over $\mathbb{N}$ without any partition.

Now, for every term $u : T$, define a term ${}^*u$ of type ${}^*T$ by induction as follows

$$\begin{aligned} {}^*0 &:= \lambda m^{\mathtt{N}}\ 0 \\ {}^*\mathtt{True} &:= \lambda m^{\mathtt{N}}\ \mathtt{True} \\ {}^*\mathtt{False} &:= \lambda m^{\mathtt{N}}\ \mathtt{False} \\ {}^*\mathsf{S} &:= \lambda f^{{}^*\mathtt{N}} \lambda m^{\mathtt{N}}\ \mathsf{S}(f(m)) \\ {}^*\mathsf{if} &:= \lambda g^{{}^*\mathtt{Bool}} \lambda f_1^{{}^*\mathtt{N}} \lambda f_2^{{}^*\mathtt{N}} \lambda m^{\mathtt{N}}\ \mathsf{if}\, f(m) f_1(m) f_2(m) \\ {}^*\mathsf{R} &:= \lambda f_1^{{}^*\mathtt{N}} \lambda f_2^{{}^*\mathtt{N} \to {}^*\mathtt{N} \to {}^*\mathtt{N}} \lambda g^{{}^*\mathtt{N}} \lambda m^{\mathtt{N}}\ (\mathsf{R}_B f_1 (\lambda n^{\mathtt{N}} f_2 (\lambda x^{\mathtt{N}} n)) g(m))(m) \\ {}^*(x^A) &:= x^{{}^*A} \\ {}^*(ut) &:= {}^*u {}^*t \\ {}^*(\lambda x^A u) &:= \lambda x^{{}^*A} {}^*u \\ {}^*\langle u, t \rangle &:= \langle {}^*u, {}^*t \rangle \\ {}^*(\pi_i u) &:= \pi_i {}^*u \end{aligned}$$

with the type B of R_B equal to ${}^*\mathtt{N} \to (\mathtt{N} \to {}^*\mathtt{N} \to {}^*\mathtt{N}) \to \mathtt{N} \to {}^*\mathtt{N}$.

The definition of the constants and the functions ${}^*\mathsf{S}$ and ${}^*\mathsf{if}$ is exactly the one used in the construction of ultrapower models of natural numbers. The definition of ${}^*\mathsf{R}$ is different because involves higher type arguments, but it is a straightforward generalization of the ultrapower construction. Intuitively, ${}^*\mathsf{R} f_1 f_2 g$ has to iterate f_2 a number of times given by g. But since g is now an hypernatural number, the concept "g times" makes no direct sense. Hence, ${}^*\mathsf{R}$ also picks as input a number m, transform g into $g(m)$ and iterates f_2 a number of times given by $g(m)$. But since f_2 is of type ${}^*\mathtt{N} \to {}^*\mathtt{N} \to {}^*\mathtt{N}$, the function given to R_B is not directly f_2, but a term $\lambda n^{\mathtt{N}} f_2(\lambda x^{\mathtt{N}} n)$ that transforms n into its hypernatural counterpart and gives it as the first argument of f_2. After all this work is done, one obtains an hypernatural

$$h := \mathsf{R}_B f_1(\lambda n^{\mathtt{N}} f_2(\lambda x^{\mathtt{N}} n)) g(m)$$

So if ${}^*\mathsf{R}$ stopped here, it would not return the right type of object. Hence, it returns $h(m)$, consistently to the fact that g has been instantiated to m previously.

The above construction can be generalized to Gödel's T, with a little more effort to be put in the generalization of ${}^*\mathsf{R}$ and ${}^*\mathsf{if}$ to all types. A version of $\mathcal{T}$ just manipulating hypernaturals is not enough for our purposes, and will be included in our final construction, so details are postponed to the next sections.

4.3. Interpretation of $\mathcal{T}$ in the Model of Hypernaturals with Moduli of Convergence

In the context of this work, we are not interested into the whole collection of hypernatural numbers, but only in those who are *convergent*. Moreover, we want also to produce, for each one of these convergent hypernaturals, a modulus of convergence. The idea therefore is to put more constructive information into the model of hypernatural numbers and to define operations that preserve this information. The new objects we are going to consider are *hypernatural numbers with moduli of convergence*. They can be represented as pairs

$$\langle \mathcal{N}, f \rangle$$

where $f : \mathbb{N} \to \mathbb{N}$ is an hypernatural number and $\mathcal{N} : (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})$ is modulus of convergence for f, as in definition 5. The resulting model is the full type structure generated as usual over these basic objects and their equivalent in the other atomic types by interpreting $\to$ as the function space constructor and $\times$ as the cartesian product. We will call it the *model of hypernaturals with moduli*. In the following, for any $s \in$w.i., $[\![u]\!]_s$ will be the denotation of a term u of $\mathcal{T}$ in this new model and the aim of this sections is to syntactically define the interpretation function $[\![\_]\!]_s$.

In order to construct such a model, we will have to define new operations that, first, generalize the ones over hypernaturals we have previously studied and, secondly, are also able to combine moduli of convergence.

For example, how to define the non standard version $[\![+]\!]_s$ of addition? The summands are two objects of the form $\langle \mathcal{N}_1, f_1 \rangle$ and $\langle \mathcal{N}_2, f_2 \rangle$. The second component of the sum will be the non standard sum

$$f_1 {}^*\!+ f_2 := \lambda m^{\mathtt{N}} f_1(m) + f_2(m)$$

of f_1 and f_2. The first component will be a modulus of convergence for $f_1 {}^*\!+ f_2$, and so a simultaneous modulus of convergence for both f_1 and f_2 is enough. From proposition 1, we know how to compute it with $\sqcup$ from $\mathcal{N}_1$ and $\mathcal{N}_2$. We can thus define

$$[\![+]\!]_s \langle \mathcal{N}_1, f_1 \rangle \langle \mathcal{N}_2, f_2 \rangle := \langle \mathcal{N}_1 \sqcup \mathcal{N}_2, f_1 {}^*\!+ f_2 \rangle$$

We now launch into the definition of our syntactically described model for the whole system $\mathcal{T}$. First we define the intended interpretation M_T of every type T.

Definition 7 (Interpretation of Types). *For every type T of system $\mathcal{T}$, we define a type M_T by induction on T as follows.*

1. *$T = A$, with A atomic. Then*

$$\mathsf{M}_A := ((\mathtt{N} \to \mathtt{N}) \to (\mathtt{N} \to \mathtt{N})) \times (\mathtt{N} \to A)$$

2. *$T = A \to B$. Then*

$$\mathsf{M}_{A\to B} := \mathsf{M}_A \to \mathsf{M}_B$$

3. *$T = A \times B$. Then*

$$\mathsf{M}_{A\times B} := \mathsf{M}_A \times \mathsf{M}_B$$

If A is atomic, the interpretation M_A of A is the set of pairs whose second component is a function $\mathtt{N} \to A$ and the first is its modulus of convergence. This is an accord with our view that whenever a $s \in$w.i. is fixed, a term t of atomic type can be interpreted as a function $\lambda m^{\mathtt{N}} t[s_m]$ paired with a modulus of convergence. The intended model of hypernaturals with moduli can be seen as the collection of sets denoted by types M_T, for T varying on all types of $\mathcal{T}$.

We now define a logical relation between the terms of our intended model of hypernaturals with moduli and the terms of system $\mathcal{T}$. It formally states what properties any denotation of any term of $\mathcal{T}$ should have. It formalizes of our previous description of what the model should contain.

Definition 8 (Generalized Modulus of Convergence). *Let t and $\mathcal{M}$ be closed terms of $\mathcal{T}$ and $s \in$w.i.. We define the relation $\mathcal{M}$ gmc_s t - representing the notion "$\mathcal{M}$ is a* generalized modulus of convergence *for t" - by induction on the type T of t as follows:*

1. *$T = A$, with A atomic. Let $\mathcal{M} : \mathsf{M}_A$. Then*

   $$\mathcal{M}\ \mathsf{gmc}_s\ t \iff \mathcal{M} = \langle \mathcal{L}, g\rangle,\ \mathcal{L} \textit{ is a modulus of convergence for } g \textit{ and } g \overset{ext}{=} \lambda n^{\mathtt{N}} t[s_n]$$

   where we have defined $(g \overset{ext}{=} \lambda n^{\mathtt{N}} t[s_n]) \equiv$ for all numerals m, $g(m) = t[s_m]$.

2. *$T = A \to B$. Let $\mathcal{M} : \mathsf{M}_{A\to B}$. Then*

   $$\mathcal{M}\ \mathsf{gmc}_s\ t \iff (\forall u^A.\ \mathcal{N}\ \mathsf{gmc}_s\ u \implies \mathcal{M}\mathcal{N}\ \mathsf{gmc}_s\ tu)$$

3. *$T = A \times B$. Let $\mathcal{M} : \mathsf{M}_{A\times B}$. Then*

   $$\mathcal{M}\ \mathsf{gmc}_s\ t \iff (\pi_0\mathcal{M}\ \mathsf{gmc}_s\ \pi_0 t \wedge \pi_1\mathcal{M}\ \mathsf{gmc}_s\ \pi_1 t)$$

The aim of the rest of this section is to syntactically define a semantic interpretation $[\![\_]\!]_s$ of the terms of $\mathcal{T}$ into the model of hypernaturals with moduli, such that for every term $u : A$ and $s \in$w.i., $[\![u]\!]_s$ gmc_s u. This means that, if A is atomic, u is evaluated in a pair $\langle \mathcal{L}, g\rangle$ such that $g \overset{\text{ext}}{=} \lambda n^{\mathtt{N}} u[s_n]$ and $\mathcal{L}$ is a modulus of convergence of g. Then, given any term $t : (\mathtt{N} \to \mathtt{N}) \to A$ of $\mathcal{T}$, if we set $u := t\Phi$ and consider $[\![u]\!]_s$, we automatically obtain a constructive proof of theorem 5.

In the following, we will make repeated use of the fact that the notion of generalized modulus of convergence is consistent with respect to equality.

Lemma 6 (Equality Soundness). *Suppose* $\mathcal{M}_1$ gmc_s t_1*,* $\mathcal{M}_1 = \mathcal{M}_2$ *and* $t_1 = t_2$*. Then* $\mathcal{M}_2$ gmc_s t_2*.*

Proof. Trivial induction on the type of T.

We now define a fundamental operation on moduli of convergence. The construction is a generalization of the one in proposition 2.

Definition 9 (Collection of Moduli Turned into a Single Modulus). *Let* $\mathcal{N} : A \to \mathsf{M}_T$ *and* $\langle \mathcal{M}, g\rangle : \mathsf{M}_A$*, with* A *atomic. We define by induction on* T *and by cases a term* $\mathcal{H}(\langle \mathcal{M}, g\rangle, \mathcal{N})$ *of type* M_T*.*

1. *T atomic. Then*

$$\mathcal{H}(\langle \mathcal{M}, g\rangle, \mathcal{N}) := \langle \mathcal{H}_1(\mathcal{M}, \lambda a^A \pi_0(\mathcal{N}_a), g),\ \lambda n^{\mathtt{N}} f_{g(n)}(n)\rangle$$

with $f := \lambda a^A \pi_1 \mathcal{N}_a$ *and* $\mathcal{H}_1$ *as in proposition 2.*

2. *$T = C \to B$. Then*

$$\mathcal{H}(\langle \mathcal{M}, g\rangle, \mathcal{N}) := \lambda \mathcal{L}^{\mathsf{M}_C} \mathcal{H}(\langle \mathcal{M}, g\rangle, \lambda a^A \mathcal{N}_a \mathcal{L})$$

3. *$T = C \times B$. Then*

$$\mathcal{H}(\langle \mathcal{M}, g\rangle, \mathcal{N}) := \langle \mathcal{H}(\langle \mathcal{M}, g\rangle, \lambda a^A \pi_0 \mathcal{N}_a), \mathcal{H}(\langle \mathcal{M}, g\rangle, \lambda a^A \pi_1 \mathcal{N}_a)\rangle$$

If we call "object of type A" any closed normal term of type A, then the role of the term $\mathcal{H}$ is to satisfy the following lemma, which is one the most important pieces of our construction. It provides a way of constructing the semantics of a term ut, with t of atomic type A, if one is able to define a semantics for t and for ua for every object a of type A.

Lemma 7. *Let* u *and* t *be closed terms respectively of types* $A \to T$ *and* A*, with* A *atomic. Suppose that for every object* a *of type* A*,* $\mathcal{N}_a$ gmc_s ua *and* $\langle \mathcal{M}, g\rangle$ gmc_s t*. Then* $\mathcal{H}(\langle \mathcal{M}, g\rangle, \mathcal{N})$ gmc_s ut*.*

Proof. By induction on T and by cases.

1. T atomic. By definition 9, we have

$$\mathcal{H}(\langle \mathcal{M}, g\rangle, \mathcal{N}) := \langle \mathcal{H}_1(\mathcal{M}, \lambda a^A \pi_0(\mathcal{N}_a), g), \lambda n^{\mathtt{N}} f_{g(n)}(n)\rangle$$

with

$$f := \lambda a^A \pi_1 \mathcal{N}_a$$

and

$$g \overset{\text{ext}}{=} \lambda n^{\mathtt{N}} t[s_n]$$

for by hypothesis $\langle \mathcal{M}, g\rangle$ gmc_s t. Moreover, for every object a of type A

$$f_a \overset{\text{ext}}{=} \lambda n^{\mathtt{N}} ua[s_n]$$

since by hypothesis $\mathcal{N}_a$ gmc_s ua. We must show that

$$\mathcal{H}_1(\mathcal{M}, \lambda a^A \pi_0 \mathcal{N}_a, g)$$

is a modulus of convergence for the function $\lambda n^{\mathtt{N}} f_{g(n)}(n)$ and that $\lambda n^{\mathtt{N}} f_{g(n)} \stackrel{\text{ext}}{=} \lambda n^{\mathtt{N}} ut[s_n]$. For this last part, indeed, for every numeral m, there is an object $a = g(m)$ such that

$$\begin{aligned}(\lambda n^{\mathtt{N}} f_{g(n)}(n))m &= f_a(m) \\ &\stackrel{\text{ext}}{=} u[s_m](a) \\ &= u[s_m](g(m)) \\ &\stackrel{\text{ext}}{=} u[s_m]((\lambda n^{\mathtt{N}} t[s_n])m) \\ &= u[s_m](t[s_m]) \\ &= (\lambda n^{\mathtt{N}} ut[s_n])m\end{aligned}$$

Now, since $\langle \mathcal{M}, g \rangle$ gmc_s t, $\mathcal{M}$ is a modulus of convergence for g. Moreover, for every object a of type A, $\mathcal{N}_a$ gmc_s ua by hypothesis, and therefore $\pi_0 \mathcal{N}_a$ is a modulus of convergence for $\lambda n^{\mathtt{N}}\ ua[s_n] \stackrel{\text{ext}}{=} f_a$. By proposition 2, we obtain that

$$\mathcal{H}_1(\mathcal{M}, \lambda a^A \pi_0 \mathcal{N}_a, g)$$

is modulus of convergence for $\lambda n^{\mathtt{N}} f_{g(n)}(n)$, and we are done.

2. $T = C \to B$. Let $v : C$ and suppose $\mathcal{L}$ gmc_s v. We have to show that

$$\mathcal{H}(\langle \mathcal{M}, g \rangle, \mathcal{N})\mathcal{L} = \mathcal{H}(\langle \mathcal{M}, g \rangle, \lambda a^A \mathcal{N}_a \mathcal{L})\ \mathsf{gmc}_s\ utv$$

But for every object a of type A, $\mathcal{N}_a$ gmc_s ua. Therefore, for every object a of type A

$$\mathcal{N}_a \mathcal{L}\ \mathsf{gmc}_s\ uav = (\lambda m^A umv)a$$

By induction hypothesis

$$\mathcal{H}(\langle \mathcal{M}, g \rangle, \lambda a^A \mathcal{N}_a \mathcal{L})\ \mathsf{gmc}_s\ (\lambda m^A umv)t = utv$$

which is the thesis.

3. $T = C \times B$. We have to show that, for $i = 0, 1$,

$$\begin{aligned}\pi_i \mathcal{H}(\langle \mathcal{M}, g \rangle, \mathcal{N}) &= \pi_i \langle \mathcal{H}(\langle \mathcal{M}, g \rangle, \lambda a^A \pi_0 \mathcal{N}_a), \mathcal{H}(\langle \mathcal{M}, g \rangle, \lambda a^A \pi_1 \mathcal{N}_a) \rangle \\ &= \mathcal{H}(\langle \mathcal{M}, g \rangle, \lambda a^A \pi_i \mathcal{N}_a) \rangle\ \mathsf{gmc}_s\ \pi_i(ut)\end{aligned}$$

Now, for every object a of type A, $\mathcal{N}_a$ gmc_s ua. Therefore, for every object a of type A

$$\pi_i \mathcal{N}_a\ \mathsf{gmc}_s\ \pi_i(ua) = (\lambda m^A \pi_i(um))a$$

By induction hypothesis

$$\mathcal{H}(\langle \mathcal{M}, g \rangle, \lambda a^A \pi_i \mathcal{N}_a)\ \mathsf{gmc}_s\ (\lambda m^A \pi_i(um))t = \pi_i(ut)$$

which is the thesis.

We are now in a position to define for each constant c of $\mathcal{T}$ a term $[\![c]\!]_s$, which is intended to satisfy the relation $[\![c]\!]_s \ \mathsf{gmc}_s \ c$. $[\![c]\!]_s$ can be seen as the non standard version of the operation denoted by c.

Definition 10 (Generalized Moduli of Convergence for Constants). *We define for every constant $c : T$ and $s \in$ w.i. a closed term $[\![c]\!]_s : \mathsf{M}_T$, accordingly to the form of c.*

1. $c : A$, *A atomic. For any closed term u of atomic type, define*

$$\mathcal{M}_{\mathsf{id},u} := \langle \lambda h^{\mathtt{N}\to\mathtt{N}} \lambda m^{\mathtt{N}} m, \lambda n^{\mathtt{N}} u \rangle$$

Then

$$[\![c]\!]_s := \mathcal{M}_{\mathsf{id},c}$$

2. $c = \Phi : \mathtt{N} \to \mathtt{N}$. *Let*

$$\mathcal{N} := \lambda n^{\mathtt{N}} \langle \lambda h^{\mathtt{N}\to\mathtt{N}} \lambda m^{\mathtt{N}} \mathsf{if}\ s_m(n) = s_{h(m)}(n)\ \mathsf{then}\ m\ \mathsf{else}\ h(m), \lambda m^{\mathtt{N}}\ s_m(n) \rangle$$

Then

$$[\![\Phi]\!]_s := \lambda \langle \mathcal{M}, g \rangle^{\mathsf{M}_{\mathtt{N}}} \mathcal{H}(\langle \mathcal{M}, g \rangle, \mathcal{N})$$

3. $c \neq \Phi, c \neq \mathsf{if}$, $c : A_0 \to \cdots \to A_m \to A$, *with A, A_i atomic for $i = 0, \ldots, m$. If $m > 0$, then define*

$$[\![c]\!]_s := \lambda \langle \mathcal{L}_0, g_0 \rangle^{\mathsf{M}_{A_0}} \ldots \lambda \langle \mathcal{L}_m, g_m \rangle^{\mathsf{M}_{A_m}} \langle \mathcal{L}_0 \sqcup \mathcal{L}_1 \sqcup \ldots \sqcup \mathcal{L}_m, \lambda n^{\mathtt{N}} c(g_0(n)) \ldots (g_m(n)) \rangle$$

assuming left association for $\sqcup$.
If $m = 0$ ($c : A_0 \to A$), define

$$[\![c]\!]_s := \lambda \langle \mathcal{M}, g \rangle^{\mathsf{M}_{A_0}} \langle \mathcal{M}, \lambda n^{\mathtt{N}} c(g(n)) \rangle$$

4. $c = \mathsf{R}_T$, *R_T recursor constant with $T = A \to (\mathtt{N} \to A \to A) \to \mathtt{N} \to A$. Define*

$$\mathcal{N} := \lambda n^{\mathtt{N}}\ \mathsf{R}_U \mathcal{I}(\lambda n^{\mathtt{N}} \mathcal{L} \mathcal{M}_{\mathsf{id},n}) n$$

with

$$U := \mathsf{M}_A \to (\mathtt{N} \to \mathsf{M}_A \to \mathsf{M}_A) \to \mathtt{N} \to \mathsf{M}_A$$

Then

$$[\![\mathsf{R}_T]\!]_s := \lambda \mathcal{I}^{\mathsf{M}_A} \lambda \mathcal{L}^{\mathsf{M}_{\mathtt{N}\to A\to A}} \lambda \langle \mathcal{M}, g \rangle^{\mathsf{M}_{\mathtt{N}}} \mathcal{H}(\langle \mathcal{M}, g \rangle, \mathcal{N})$$

5. $c = \mathsf{if}_T$ *with $T : \mathtt{Bool} \to A \to A \to A$. Define*

$$\mathcal{N} := \lambda b^{\mathtt{Bool}} \mathsf{if}\ b\ \mathsf{then}\ \mathcal{L}_1\ \mathsf{else}\ \mathcal{L}_2,$$

Then

$$[\![\mathsf{if}_T]\!]_s := \lambda \langle \mathcal{M}, g \rangle^{\mathsf{M}_{\mathtt{Bool}}} \lambda \mathcal{L}_1^{\mathsf{M}_A} \lambda \mathcal{L}_2^{\mathsf{M}_A} \mathcal{H}(\langle \mathcal{M}, g \rangle, \mathcal{N})$$

The definition of $[\![c]\!]_s$ is a generalization of the operations done with hypernaturals. We remark that only the interpretation of the constant Φ depends on s.

In case (1), we transform basic objects into their hypernatural, hyperboolean and hyperconstant counterparts (we call them hyperobjects) all paired with their trivial moduli of convergence.

In case (2), the interpretation $[\![\Phi]\!]_s$ of Φ is obtained by first defining uniformly on the numeral parameter n a collection of interpretations $\mathcal{N}_n = [\![\Phi n]\!]_s$ of Φn, and then using the term $\mathcal{H}$ to put together the interpretations in order to define $[\![\Phi]\!]_s\langle\mathcal{M}, g\rangle$.

In case (3), we provide the non standard version of the function represented by c, which is a function $[\![c]\!]_s$ which combines both hyperobjects and their moduli of convergence.

In case (4) and (5) we have generalized the ideas of subsection 4.2. In particular, for $T = A \to (\mathtt{N} \to A \to A) \to \mathtt{N} \to A$ and A atomic, the definition of $[\![\mathsf{R}_T]\!]_s$ is exactly the same of ${}^*\mathsf{R}$ in subsection 4.2, enriched with the information of how to combine moduli of convergence. In fact, if we consider the term

$$\begin{aligned}&[\![\mathsf{R}_T]\!]_s \mathcal{I}\mathcal{L}\langle\mathcal{M}, g\rangle \\ =&\langle\mathcal{H}_1(\mathcal{M}, \lambda n^{\mathtt{N}}\pi_0(\mathcal{N}_n), g),\ \lambda n^{\mathtt{N}} f_{g(n)}(n)\rangle\end{aligned}$$

with $f := \lambda n^{\mathtt{N}}\pi_1(\mathcal{N}_n)$, its right projection is equal to

$$\lambda n^{\mathtt{N}} f_{g(n)}(n)$$

which is equal to

$$\lambda n^{\mathtt{N}}(\pi_1 \mathsf{R}_U \mathcal{I}(\lambda n^{\mathtt{N}}\mathcal{L}\mathcal{M}_{\mathsf{id},n})g(n))(n)$$

which corresponds exactly to the term

$${}^*\mathsf{R} f_1 f_2 g = \lambda n^{\mathtt{N}}\ (\mathsf{R}_B f_1(\lambda n^{\mathtt{N}} f_2(\lambda x^{\mathtt{N}} n))g(n))(n)$$

of subsection 4.2.

We now prove that for any constant c, $[\![c]\!]_s$ is a generalized modulus of convergence for c.

Proposition 3. *For every constant c, $[\![c]\!]_s$ gmc_s c.*

Proof. We proceed by cases, accordingly to the form of c.

1. $c = \Phi$. Let $t : \mathtt{N}$ and suppose $\langle\mathcal{M}, g\rangle$ gmc_s t. We have to prove that

   $$[\![\Phi]\!]_s\langle\mathcal{M}, g\rangle\ \mathsf{gmc}_s\ \Phi t$$

   By definition 10 of $[\![\Phi]\!]_s$

   $$[\![\Phi]\!]_s\langle\mathcal{M}, g\rangle = \mathcal{H}(\langle\mathcal{M}, g\rangle, \mathcal{N})$$

   with

   $$\mathcal{N} := \lambda n^{\mathtt{N}}\langle\lambda h^{\mathtt{N}\to\mathtt{N}}\lambda m^{\mathtt{N}}\mathsf{if}\ s_m(n) = s_{h(m)}(n)\ \mathsf{then}\ m\ \mathsf{else}\ h(m), \lambda m^{\mathtt{N}} s_m(n)\rangle$$

Since $\langle \mathcal{M}, g\rangle$ gmc_s t, if we prove that for every numeral n, $\mathcal{N}_n$ gmc_s Φn, we obtain by lemma 7 that $\mathcal{H}(\langle \mathcal{M}, g\rangle, \mathcal{N})$ gmc_s Φt and we are done. So let us show that, given a numeral n, $\pi_0\mathcal{N}_n$ is a modulus of convergence for the function

$$\pi_1\mathcal{N}_n = \lambda m^{\mathtt{N}} s_m(n) \overset{\text{ext}}{=} \lambda m^{\mathtt{N}}\ \Phi(n)[s_m]$$

We have to prove that given any closed term $h^{\mathtt{N}\to\mathtt{N}} \geq \mathsf{id}$ and numeral n,

$$\lambda m^{\mathtt{N}} s_m(n) \downarrow [(\pi_0\mathcal{N}_n)_h(z), h((\pi_0\mathcal{N}_n)_h(z))]$$

We have two possibilities:
i) $s_z(n) = s_{h(z)}(n)$. Since $s \in$ w.i., we have either $s_{h(z)}(n) = 0$ and so

$$\forall y^{\mathtt{N}}.\ z \leq y \leq h(z) \implies s_y(n) = 0$$

or $s_z(n) = s_{h(z)}(n) \neq 0$ and so

$$\forall y^{\mathtt{N}}.\ z \leq y \leq h(z) \implies s_y(n) = s_z(n)$$

Therefore

$$\begin{aligned} &\lambda m^{\mathtt{N}} s_m(n) \downarrow [z, h(z)] \\ &\qquad = [(\pi_0\mathcal{N}_n)_h(z), h((\pi_0(\mathcal{N}_n)_h(z))] \end{aligned}$$

by definition of $\mathcal{N}$.
ii) $s_z(n) \neq s_{h(z)}(n)$. Since $s \in$ w.i., we have $s_z \preccurlyeq s_{h(z)}$ and hence $0 = s_z(n)$. So $s_{h(z)}(n) = s_{h(h(z))}(n)$ and as above

$$\begin{aligned} &\lambda m^{\mathtt{N}} s_m(n) \downarrow [h(z), h(h(z))] \\ &\qquad = [(\pi_0\mathcal{N}_n)_h(z), h((\pi_0\mathcal{N}_n)_h(z))] \end{aligned}$$

2. $c \neq \Phi$, $c \neq \mathsf{if}$, $c : A_0 \to \cdots \to A_m \to A$.
i) $m > 0$. Suppose $t_i : A_i$ and $\langle \mathcal{L}_i, g_i\rangle$ gmc_s t_i for all $i = 0, \ldots, m$. We have to prove that

$$[\![c]\!]_s \langle \mathcal{L}_0, g_0\rangle \ldots \langle \mathcal{L}_m, g_m\rangle\ \mathsf{gmc}_s\ ct_1 \ldots t_m$$

We have that $g_i \overset{\text{ext}}{=} \lambda n^{\mathtt{N}} t_i[s_n]$ for $i = 0, \ldots, m$. Moreover, since by definition 10 of $[\![c]\!]_s$

$$\begin{aligned} &[\![c]\!]_s \langle \mathcal{L}_0, g_0\rangle \ldots \langle \mathcal{L}_m, g_m\rangle \\ &= \langle \mathcal{L}_0 \sqcup \mathcal{L}_1 \sqcup \ldots \sqcup \mathcal{L}_m, \lambda n^{\mathtt{N}} c(g_0(n)) \ldots (g_m(n))\rangle \end{aligned}$$

we must show that

$$\begin{aligned} &\pi_0([\![c]\!]_s \langle \mathcal{L}_0, g_0\rangle \ldots \langle \mathcal{L}_m, g_m\rangle) \\ &= \mathcal{L}_0 \sqcup \mathcal{L}_1 \sqcup \ldots \sqcup \mathcal{L}_m \end{aligned}$$

is a modulus of convergence for

$$\begin{aligned} &\lambda n^{\mathtt{N}} c(g_0(n)) \ldots (g_m(n)) \\ &\overset{\text{ext}}{=} \lambda n^{\mathtt{N}} c(t_1[s_n]) \ldots (t_m[s_n]) \\ &= \lambda n^{\mathtt{N}} ct_1 \ldots t_m[s_n] \end{aligned}$$

Since for $i = 0, \ldots, m$, $\mathcal{L}_i$ is a modulus of convergence for g_i, by repeated application of proposition 1 we deduce that $\mathcal{L}_0 \sqcup \mathcal{L}_1 \sqcup \ldots \sqcup \mathcal{L}_m$ is modulus of convergence for all $g_1, \ldots, g_m$ simultaneously. Hence for all closed terms $h : \mathtt{N} \to \mathtt{N} \geq \mathsf{id}$ and numerals z, and for $i = 0, \ldots, m$

$$g_i \downarrow [(\mathcal{L}_0 \sqcup \mathcal{L}_1 \sqcup \ldots \sqcup \mathcal{L}_m)_h(z), h((\mathcal{L}_0 \sqcup \mathcal{L}_1 \sqcup \ldots \sqcup \mathcal{L}_m)_h(z))]$$

and therefore

$$\lambda m^{\mathtt{N}}\ c(g_1(m)) \ldots (g_n(m)) \downarrow [(\mathcal{L}_0 \sqcup \mathcal{L}_1 \sqcup \ldots \sqcup \mathcal{L}_m)_h(z), h((\mathcal{L}_0 \sqcup \mathcal{L}_1 \sqcup \ldots \sqcup \mathcal{L}_m)_h(z))]$$

which is the thesis.
ii) $m = 0$. Straightforward simplification of the argument for i).

3. $c : A$, A atomic. By definition 10

$$[\![c]\!]_s = \langle \lambda h^{\mathtt{N}\to\mathtt{N}} \lambda m^{\mathtt{N}} m, \lambda n^{\mathtt{N}} c \rangle$$

We have therefore to prove that $\lambda h^{\mathtt{N}\to\mathtt{N}} \lambda m^{\mathtt{N}} m$ is a modulus of convergence for $\lambda n^{\mathtt{N}} c$, which is trivially true, and that $\lambda n^{\mathtt{N}} c[s_n] = \lambda n^{\mathtt{N}} c$, which is also trivial. We conclude $[\![c]\!]_s\ \mathsf{gmc}_s\ c$.

4. $c = \mathsf{R}_T$, R_T recursor constant with $T = A \to (\mathtt{N} \to A \to A) \to \mathtt{N} \to A$. Suppose $\mathcal{I}\ \mathsf{gmc}_s\ u : A$, $\mathcal{L}\ \mathsf{gmc}_s\ v : \mathtt{N} \to A \to A$ and $\langle \mathcal{M}, g \rangle\ \mathsf{gmc}_s\ t : \mathtt{N}$. We have to prove that

$$[\![\mathsf{R}_T]\!]_s \mathcal{I}\mathcal{L}\langle \mathcal{M}, g \rangle = \mathcal{H}(\langle \mathcal{M}, g \rangle, \mathcal{N})\ \mathsf{gmc}_s\ \mathsf{R}_T uvt$$

where

$$\mathcal{N} := \lambda n^{\mathtt{N}}\ \mathsf{R}_U \mathcal{I}(\lambda n^{\mathtt{N}} \mathcal{L}\mathcal{M}_{\mathsf{id},n})n$$

If we show that for all numerals n, $\mathcal{N}_n\ \mathsf{gmc}_s\ \mathsf{R}_T uvn$, by lemma 7 we obtain that

$$\mathcal{H}(\langle \mathcal{M}, g \rangle, \mathcal{N})\ \mathsf{gmc}_s\ \mathsf{R}_T uvt$$

We prove that by induction on n.
If $n = 0$, then

$$\mathcal{N}_0 = \mathsf{R}_U \mathcal{I}(\lambda n^{\mathtt{N}} \mathcal{L}\mathcal{M}_{\mathsf{id},n})0 = \mathcal{I}\ \mathsf{gmc}_s\ u = \mathsf{R}_T uv0$$

If $n = \mathsf{S}(m)$, then

$$\begin{aligned}
\mathcal{N}_{\mathsf{S}(m)} &= \mathsf{R}_U \mathcal{I}(\lambda n^{\mathtt{N}} \mathcal{L}\mathcal{M}_{\mathsf{id},n})\mathsf{S}(m) \\
&= (\lambda n^{\mathtt{N}} \mathcal{L}\mathcal{M}_{\mathsf{id},n})m(\mathsf{R}_U \mathcal{I}(\lambda n^{\mathtt{N}} \mathcal{L}\mathcal{M}_{\mathsf{id},n})m) \\
&= \mathcal{L}\mathcal{M}_{\mathsf{id},m}(\mathsf{R}_U \mathcal{I}(\lambda n^{\mathtt{N}} \mathcal{L}\mathcal{M}_{\mathsf{id},n})m) \\
&= \mathcal{L}\mathcal{M}_{\mathsf{id},m}\mathcal{N}_m
\end{aligned}$$

By induction hypothesis, $\mathcal{N}_m\ \mathsf{gmc}_s\ \mathsf{R}_T uvm$. Moreover, $\mathcal{M}_{\mathsf{id},m}\ \mathsf{gmc}_s\ m$ and by hypothesis $\mathcal{L}\ \mathsf{gmc}_s\ v$. Hence

$$\mathcal{L}\mathcal{M}_{\mathsf{id},m}\mathcal{N}_m\ \mathsf{gmc}_s\ vm(\mathsf{R}_T uvm) = \mathcal{R}_T uv\mathsf{S}(m)$$

which is the thesis.

5. $c = \mathsf{if}_T$, with $T : \texttt{Bool} \to A \to A \to A$. Suppose $\mathcal{L}_1\ \mathsf{gmc}_s\ u_1 : A$, $\mathcal{L}_2\ \mathsf{gmc}_s\ u_2 : A$ and $\langle \mathcal{M}, g\rangle\ \mathsf{gmc}_s\ t : \texttt{Bool}$. We have to prove that

$$[\![\mathsf{if}_T]\!]_s \langle \mathcal{M}, g\rangle \mathcal{L}_1 \mathcal{L}_2 = \mathcal{H}(\langle \mathcal{M}, g\rangle, \mathcal{N})\ \mathsf{gmc}_s\ \mathsf{if}_T t u_1 u_2$$

where

$$\mathcal{N} := \lambda b^{\texttt{Bool}} \mathsf{if}\ b\ \mathsf{then}\ \mathcal{L}_1\ \mathsf{else}\ \mathcal{L}_2$$

If we show that for all $a \in \{\texttt{True}, \texttt{False}\}$, $\mathcal{N}_a\ \mathsf{gmc}_s\ (\lambda b^{\texttt{Bool}} \mathsf{if}_T b u_1 u_2)a$, by lemma 7 we obtain that

$$\mathcal{H}(\langle \mathcal{M}, g\rangle, \mathcal{N})\ \mathsf{gmc}_s\ (\lambda b^A \mathsf{if}_T b u_1 u_2)t = \mathsf{if}_T t u_1 u_2$$

We prove that by cases.
If $a = \texttt{True}$, then

$$\mathcal{N}_a = \mathcal{L}_1\ \mathsf{gmc}_s\ u_1 = (\lambda b^A \mathsf{if}_T b u_1 u_2)a$$

If $a = \texttt{False}$, then

$$\mathcal{N}_a = \mathcal{L}_2\ \mathsf{gmc}_s\ u_2 = (\lambda b^A \mathsf{if}_T b u_1 u_2)a$$

Hence, we have the thesis.

We are finally ready to define the interpretation of every term of $\mathcal{T}$ in our model of hypernaturals with moduli.

Definition 11 (Generalized Moduli of Convergence for Terms of $\mathcal{T}$). *For every term $v : T$ of system $\mathcal{T}$ and $s \in$w.i., we define a term $[\![v]\!]_s : \mathsf{M}_T$ by induction on v and by cases as follows:*

1. *$v = c$, with c constant. We define $[\![c]\!]_s$ as in definition 10.*

2. *$v = x^A$, x variable. Then*
$$[\![x^A]\!]_s := x^{\mathsf{M}_A}$$

3. *$v = ut$. Then*
$$[\![ut]\!]_s := [\![u]\!]_s [\![t]\!]_s$$

4. *$v = \lambda x^A u$. Then*
$$[\![\lambda x^A u]\!]_s := \lambda x^{\mathsf{M}_A}\ [\![u]\!]_s$$

5. *$v = \langle u, t\rangle$. Then*
$$[\![\langle u, t\rangle]\!]_s := \langle [\![u]\!]_s, [\![t]\!]_s\rangle$$

6. *$v = \pi_i u$. Then*
$$[\![\pi_i u]\!]_s := \pi_i [\![u]\!]_s$$

5. Adequacy Theorem

We are now able to prove our main theorem. For every closed term u, $[\![u]\!]_s$ is an inhabitant of the model of hypernaturals with moduli of convergence.

Theorem 8 (Adequacy Theorem). *Let $w : A$ be a term of $\mathcal{T}$ and let $x_1^{A_1}, \ldots, x_n^{A_n}$ contain all the free variables of w. Then, for all $s \in$w.i.*

$$\lambda x_1^{\mathsf{M}_{A_1}} \ldots \lambda x_n^{\mathsf{M}_{A_n}} [\![w]\!]_s \;\mathsf{gmc}_s\; \lambda x_1^{A_1} \ldots \lambda x_n^{A_n} w$$

PROOF. Let $t_1 : A_1, \ldots, t_n : A_n$ be arbitrary terms. We have to prove that

$$\mathcal{M}_1 \;\mathsf{gmc}_s\; t_1, \ldots, \mathcal{M}_n \;\mathsf{gmc}_s\; t_n \implies [\![w]\!]_s[\mathcal{M}_1/x_1^{\mathsf{M}_{A_1}} \ldots \mathcal{M}_n/x_n^{\mathsf{M}_{A_n}}] \;\mathsf{gmc}_s\; w[t_1/x_1^{A_1} \ldots t_n/x_n^{A_n}]$$

For any term v, we set

$$\overline{v} := v[t_1/x_1^{A_1} \cdots t_n/x_n^{A_n}]$$

and

$$\overline{[\![v]\!]_s} := [\![v]\!]_s[\mathcal{M}_1/x_1^{\mathsf{M}_{A_1}} \ldots \mathcal{M}_n/x_n^{\mathsf{M}_{A_n}}]$$

With that notation, we have to prove that $\overline{[\![w]\!]_s} \;\mathsf{gmc}_s\; \overline{w}$. The proof is by induction on w and proceeds by cases, accordingly to the form of w.

1. $w = c$, with c constant. Since $[\![c]\!]_s$ is closed and c does not have free variables, by proposition 3

   $$\overline{[\![w]\!]_s} = \overline{[\![c]\!]_s} = [\![c]\!]_s \;\mathsf{gmc}_s\; c = \overline{c} = \overline{w}$$

   which is the thesis.

2. $w = x_i^{A_i}$, for some $1 \le i \le n$. Then

   $$\overline{[\![w]\!]_s} = x_i^{\mathsf{M}_{A_i}}[\mathcal{M}_1/x_1^{\mathsf{M}_{A_1}} \ldots \mathcal{M}_n/x_n^{\mathsf{M}_{A_n}}] = \mathcal{M}_i \;\mathsf{gmc}_s\; t_i = x_i^{A_i}[t_1/x_1^{A_1} \cdots t_n/x_n^{A_n}] = \overline{w}$$

   which is the thesis.

3. $w = ut$. By induction hypothesis, $\overline{[\![u]\!]_s} \;\mathsf{gmc}_s\; \overline{u}$ and $\overline{[\![t]\!]_s} \;\mathsf{gmc}_s\; \overline{t}$. So

   $$\overline{[\![ut]\!]_s} = \overline{[\![u]\!]_s}\overline{[\![t]\!]_s} \;\mathsf{gmc}_s\; \overline{u}\overline{t} = \overline{w}$$

   which is the thesis.

4. $w = \lambda x^A u$. Let $t : A$ and suppose $\mathcal{M} \;\mathsf{gmc}_s\; t$. We have to prove that $\overline{[\![w]\!]_s}\mathcal{M} \;\mathsf{gmc}_s\; \overline{w}t$. By induction hypothesis

   $$\overline{[\![\lambda x^A u]\!]_s}\mathcal{M} = (\lambda x^{\mathsf{M}_A} \overline{[\![u]\!]_s})\mathcal{M} = \overline{[\![u]\!]_s}[\mathcal{M}/x^{\mathsf{M}_A}] \;\mathsf{gmc}_s\; \overline{u}[t/x^A] = \overline{w}t$$

   which is the thesis.

5. $w = \langle u_0, u_1\rangle$. By induction hypothesis, $\overline{[\![u_0]\!]_s}\ \mathsf{gmc}_s\ \overline{u}_0$ and $\overline{[\![u_1]\!]_s}\ \mathsf{gmc}_s\ \overline{u}_1$. Therefore, for $i = 0, 1$

$$\pi_i\overline{[\![\langle u_0, u_1\rangle]\!]_s} = \pi_i\langle\overline{[\![u_0]\!]_s[\![u_1]\!]_s}\rangle = \overline{[\![u_i]\!]_s}\ \mathsf{gmc}_s\ \overline{u}_i = \pi_i\overline{w}$$

which is the thesis.

6. $w = \pi_i u$, with $i \in \{0, 1\}$. By induction hypothesis, $\overline{[\![u]\!]_s}\ \mathsf{gmc}_s\ \overline{u}$. Therefore,

$$\overline{[\![\pi_i u]\!]_s} = \pi_i\overline{[\![u]\!]_s}\ \mathsf{gmc}_s\ \pi_i\overline{u} = \overline{w}$$

which is the thesis.

6. Consequences of the Adequacy Theorem

In this section, we spell out the most interesting consequences of adeguacy theorem.

6.1. Weak Convergence Theorem

We can finally prove the constructive version of theorem 1, our main goal. The following theorem is even stronger of the previously enunciated theorem 5, because it states that one can find moduli of convergence for any uniformly defined collection of terms.

Theorem 9 (Weak Convergence Theorem for Collection of Terms). *Let $t : \mathtt{N} \to (\mathtt{N} \to \mathtt{N}) \to \mathtt{S}$ be a closed term of $\mathcal{T}$ not containing Φ, with $\mathtt{S}$ atomic type. Then we can effectively define a closed term $\mathcal{M} : \mathtt{N} \to (\mathtt{N} \to (\mathtt{N} \to \mathtt{N})) \to (\mathtt{N} \to \mathtt{N}) \to (\mathtt{N} \to \mathtt{N})$ of $\mathcal{T}$, such that for all $s : \mathtt{N} \to (\mathtt{N} \to \mathtt{N})$, $s \in$ w.i. and numerals n, $\mathcal{M}_n s$ is a modulus of convergence for $\lambda m^{\mathtt{N}} t_n(s_m)$.*

Proof. Let

$$\mathcal{M} := \lambda y^{\mathtt{N}} \lambda s^{\mathtt{N}\to(\mathtt{N}\to\mathtt{N})} \pi_0((\lambda x^{\mathsf{M}_{\mathtt{N}}}[\![t_{x^{\mathtt{N}}}\Phi]\!]_s)\mathcal{M}_{\mathsf{id},y})$$

By the adequacy theorem 8,

$$\lambda x^{\mathsf{M}_{\mathtt{N}}}[\![t_{x^{\mathtt{N}}}\Phi]\!]_s\ \mathsf{gmc}_s\ \lambda x^{\mathtt{N}} t_{x^{\mathtt{N}}}\Phi$$

Since for every numeral n, $\mathcal{M}_{\mathsf{id},n}\ \mathsf{gmc}_s\ n$, we have

$$(\lambda x^{\mathsf{M}_{\mathtt{N}}}[\![t_{x^{\mathtt{N}}}\Phi]\!]_s)\mathcal{M}_{\mathsf{id},n}\ \mathsf{gmc}_s\ t_n\Phi$$

By definition of generalized modulus of convergence and of $\mathcal{M}$, $\mathcal{M}_n s$ is a modulus of convergence for $\lambda m^{\mathtt{N}} t_n\Phi[s_m] = \lambda m^{\mathtt{N}} t_n(s_m)$.

6.2. Zeros for Unary Update Procedures

Thanks to the adequacy theorem, we are able to give a new constructive proof of Avigad's theorem for unary update procedures. Here, we give a slightly different definition of unary update procedure, since we want a little more precise description of learning. This is not a limitation, since the update procedures which are *actually* used by Avigad [4] in proving 1-consistency of PA still fall under our definition.

Intuitively, an update procedure is a functional which takes as input a function f over $\mathbb{N}$ approximating some oracle. Then, it uses that function to compute some witnesses for some provable Σ_1^0 formula of PA. Afterwards, it checks whether the result of its computation is sound. If it is not, it identifies some wrong value $f(n)$ used in the computation and it corrects it with a new one.

Definition 12 (Update Operator, Typed Update Procedures). *Fix a primitive recursive bijective coding $|\_| : (\mathbb{N}^2 \cup \{\varnothing\}) \to \mathbb{N}$ of $\varnothing$ and of pairs of natural numbers into natural numbers. Define a binary operation $\oplus$ which combine functions $f : \mathtt{N} \to \mathtt{N}$ and codes of the form $|(n, m)|$ of pairs of natural numbers and returns a function $\mathtt{N} \to \mathtt{N}$ as follows*

$$f \oplus |(m, n)| := \lambda x^{\mathtt{N}} \mathsf{if}\ x = m\ \mathsf{then}\ n\ \mathsf{else}\ f(x)$$

For convenience, define also $f \oplus |\varnothing| = f$.

A typed unary update procedure *is a term $\mathcal{U} : (\mathtt{N} \to \mathtt{N}) \to \mathtt{N}$ of Gödel's T such that the following holds:*

1. *for all closed type-$\mathtt{N} \to \mathtt{N}$ terms f and g of T, if*

$$\mathcal{U}f = |(n, m)|,\ g(n) = m \text{ and } \mathcal{U}g = |(h, l)|$$

then $h \neq n$.

If $\mathcal{U}$ is a typed unary update procedure, a zero *for $\mathcal{U}$ is a closed term $f : \mathtt{N} \to \mathtt{N}$ of T such that $\mathcal{U}f = |\varnothing|$.*

If $\mathcal{U}$ is a unary update procedure and $f : \mathtt{N} \to \mathtt{N}$ is a closed term of T approximating some oracle Φ, there a two possibilities: either f is a fine approximation and then $\mathcal{U}f = |\varnothing|$; or f is not and then $\mathcal{U}(f) = |(n, m)|$, for some numerals n, m: $\mathcal{U}$ says the function f should be updated as to output m on input n. Moreover, if $\mathcal{U}f = |(n, m)|$, one has *learned* that $\Phi(n) = m$: by definition of update procedure, if g is another candidate approximation of Φ and $g(n) = m$, then $\mathcal{U}g$ does not represent a request to modify the value of g at point n, for $\mathcal{U}g = |(h, l)|$ implies $h \neq n$.

Every unary update procedure gives rise to a learning process, i.e. a weakly increasing chain of functions.

Proposition 4 (Learning Processes from Unary Update Procedures). *Let $\mathcal{U}$ be a unary update procedure and define by recursion $s_0 := 0^{\mathtt{N} \to \mathtt{N}} := \lambda x^{\mathtt{N}} 0$ and $s_{k+1} := s_k \oplus \mathcal{U}s_k$. Then $s \in$w.i..*

Proof. Suppose $s_i(n) = m \neq 0$. We have to prove that for all j, $s_{i+j}(n) = m$. We proceed by induction on j. Suppose $j > 0$. Since $s_0 = 0^{\mathtt{N} \to \mathtt{N}}$ and $s_i(n) \neq 0$, it must be that for some $i_0 < i$

$$\mathcal{U}s_{i_0} = |(n, m)|$$

By induction hypothesis, $s_{i+j-1}(n) = m$. By definition 12 of update procedure

$$\mathcal{U}s_{i+j-1} \neq |(n,l)|$$

for all l. Since

$$s_{i+j} = s_{i+j-1} \oplus \mathcal{U}s_{i+j-1}$$

it must be that $s_{i+j}(n) = m$.

We now prove that we can compute in system T zeros for every T-definable collection of unary typed update procedures.

Theorem 10 (Zero Theorem for Collection of Unary Typed Update Procedures). *Let $\mathcal{U} : \mathtt{N} \to (\mathtt{N} \to \mathtt{N}) \to \mathtt{N}$ be a term of T such that for all numerals n, $\mathcal{U}$ is a typed unary update procedure. Then one can constructively define a closed term $\mathsf{zero} : \mathtt{N} \to (\mathtt{N} \to \mathtt{N})$ of T such that for all numerals n*

$$\mathcal{U}_n(\mathsf{zero}_n) = |\varnothing|$$

Proof. By the weak convergence theorem 9, there exists a term $\mathcal{M}$ of T such that for all numerals n and for all $s \in$w.i., $\mathcal{M}_n s$ is a modulus of convergence for $\lambda m^{\mathtt{N}}\, \mathcal{U}_n(s_m)$. Fix now a numeral n. Define by recursion a term s such that $s_0 := 0^{\mathtt{N}\to\mathtt{N}}$ and $s_{m+1} := s_m \oplus \mathcal{U}_n(s_m)$. Then $s \in$w.i. by proposition 4, since $\mathcal{U}$ is a typed unary update procedure. We have that $\mathcal{M}_n s$ is a modulus of convergence for $\lambda m^{\mathtt{N}}\, \mathcal{U}_n(s_m)$. If we choose $h := \lambda m^{\mathtt{N}} m + 1$ and set $j := (\mathcal{M}_n s)h$, we have that

$$\mathcal{U}_n(s_j) = \mathcal{U}_n(s_{j+1})$$

by definition of modulus of convergence. So let

$$\mathsf{zero}_n := s_{(\mathcal{M}_n s)h+1}$$

Then

$$\begin{aligned} s_{j+2} &= s_{j+1} \oplus \mathcal{U}_n(s_{j+1}) \\ &= (s_j \oplus \mathcal{U}_n(s_j)) \oplus \mathcal{U}_n(s_{j+1}) \\ &= (s_j \oplus \mathcal{U}_n(s_j)) \oplus \mathcal{U}_n(s_j) \\ &= s_j \oplus \mathcal{U}_n(s_j) \\ &= s_{j+1} \end{aligned}$$

and hence it must be that

$$\mathcal{U}_n(\mathsf{zero}_n) = \mathcal{U}(s_{j+1}) = |\varnothing|$$

which is the thesis.

6.3. Learning Based Realizability and Provably Total Functions of PA

We now give a definition of learning based realizability for $\mathsf{HA} + \mathsf{EM}_1$, Heyting Arithmetic plus excluded middle over Σ^0_1 formulas. Our presentation is simplified with respect to Aschieri and Berardi [2]: we leave out all the syntactic sugar used there and instead employ concepts that we have already defined, such as that of typed unary update procedure. The advantages are that we do not have to carry out any coding in T of the syntax that appears in [2] in order to prove our results and that the link of realizability with update procedures becomes immediate and elegant. Anyway, the differences with [2] are negligible and only syntactical and for full motivations for learning based realizability we refer to it. In Aschieri [1], the results of this section are proved with respect to the same formalism used in [2].

Recall now the informal explanation of the introduction. In order to realize classical principles we first introduce oracles and then learning devices that will approximate them. The constant Φ that we have been using so far just as technical device, now represents a Skolem function for the formula $\forall x^{\mathtt{N}} \forall y^{\mathtt{N}} \exists z^{\mathtt{N}}\ Txyz$, with T Kleene's predicate (see [21]), coded in system T. In fact Φ is in the same recursive degree of an oracle for the Halting problem. Using the oracle Φ one can decide, for every n, m, if $\exists y^{\mathtt{N}}\ Tnmy$ holds, since by definition of Skolem function

$$Tnm(\Phi|(n,m)|) \equiv \exists y^{\mathtt{N}} Tnmy$$

and hence is possible to give a first ineffective Kreisel-style realizability interpretation of EM_1 and hence of $\mathsf{HA} + \mathsf{EM}_1$. Here we can assume without being too restrictive that

$$\mathsf{EM}_1 := \forall x^{\mathtt{N}} \forall y^{\mathtt{N}}.\ \exists z^{\mathtt{N}} Txyz \vee \forall z^{\mathtt{N}} \neg Txyz$$

We denote with $\mathsf{T}_{\text{Class}}$ Gödel's system T plus the constant $\Phi : \mathtt{N} \rightarrow \mathtt{N}$. There is no set of computable reduction rules for the constant Φ, and therefore no set of computable reduction rules for $\mathsf{T}_{\text{Class}}$. Thus, terms of $\mathsf{T}_{\text{Class}}$ will be computed with respect approximations of Φ: given any term $\sigma : \mathtt{N} \rightarrow \mathtt{N}$ of system T, $t[\sigma] = t[\sigma/\Phi]$ is again a term of system T and hence computable. $t[\sigma]$ is said to be an *approximation of t at σ*. The particular terms of type $\mathtt{N} \rightarrow \mathtt{N}$ we will use to approximate Φ are called states of knowledge.

Definition 13 (States of Knowledge). *A closed term $\sigma : \mathtt{N} \rightarrow \mathtt{N}$ of Gödel's T is said to be a* state of knowledge *(shortly, a state) if*

$$\forall n^{\mathtt{N}}, m^{\mathtt{N}}.\ s(|(n,m)|) \neq 0 \implies Tnms(|n,m|)$$

If s is a state, all its non "default" values (values different from zeros) are witnesses for the Kleene's predicate. A state represents thus a partial amount of information about Φ and is allowed to have default values. The choice of zero as a default value is really arbitrary and for the presentation of this paper we could have called a state of knowledge any term of type $\mathtt{N} \rightarrow \mathtt{N}$. We chose however to remain consistent with the definition in [2], which has technical advantages and is semantically clearer.

In the following, the type $\mathtt{N}$ will be used by atomic realizers - which will be update procedures - as a code of $\mathbb{N}^2 \cup \{\varnothing\}$: in order to stress when the type $\mathtt{N}$ is intended to be used like that, we will use the symbol $\mathtt{S}$ and we define $\mathtt{S} := \mathtt{N}$.

We now fix a language for Peano Arithmetic and then formulate a realizability relation between terms of $\mathsf{T}_{\text{Class}}$ and formulas of the language.

Definition 14 (The language $\mathcal{L}$ of Peano Arithmetic). *Define $\mathcal{L}$ as follows:*

1. *The terms of $\mathcal{L}$ are all terms $t \in \mathsf{T}$, such that $t : \mathtt{N}$ and the free variables of t are contained in $\{x_1^{\mathtt{N}}, \ldots, x_n^{\mathtt{N}}\}$ for some $x_1, \ldots, x_n$.*

2. *The atomic formulas of* $\mathcal{L}$ *are all terms* $Qt_1 \dots t_n \in \mathsf{T}$, *for some* $Q : \mathtt{N}^n \to \mathtt{Bool}$ closed term of T, *and some terms* $t_1, \dots, t_n$ *of* $\mathcal{L}$. *For every* $Q : \mathtt{N}^n \to \mathtt{Bool}$, *we denote with* $\neg Q : \mathtt{N}^n \to \mathtt{Bool}$ *its boolean negation (which of course is itself a term of* T*).*

3. *The formulas of* $\mathcal{L}$ *are built from atomic formulas of* $\mathcal{L}$ *by the connectives* $\vee, \wedge, \to \forall, \exists$ *as usual.*

So far in this paper, we have considered only the abstract definition of update procedure and have not seen any instance of the concept. Now we define the notion of EM_1-update procedure, which is the kind of typed unary update procedure that we need to realize EM_1.

Definition 15 (EM_1-Update Procedure). *Let* $t : (\mathtt{N} \to \mathtt{N}) \to \mathtt{S}$ *be a term of* $\mathsf{T}_{\mathit{Class}}$. t *is said to be a* EM_1-update procedure *if for all terms* $f : \mathtt{N} \to \mathtt{N}$ *of* T, *if*

$$t(f) = |(i, l)|$$

then

1. $i = |(n, m)|$, *for some* n, m.

2. $Tnmf(|(n, m)|) = \mathtt{False}$.

3. $Tnml = \mathtt{True}$.

If t is a EM_1-update procedure, then t corrects only wrong values of its input function f (condition (2)), that is, values that falsify the assertion that f approximates a Skolem function for the formula $\forall x^{\mathtt{N}}, y^{\mathtt{N}}\ \exists z^{\mathtt{N}}\ Txyz$. Moreover, the update $t(f)$ must contain a right value for f, that is a witness for the formula $\exists z^{\mathtt{N}}\ Tnmz$ (condition (3)). It is easy to see that our terminology is sound and that every EM_1-update procedure is indeed a unary update procedure.

We now define the types in which formulas are translated and that will be the types of corresponding realizers. Note that the type for atomic formula is $\mathtt{S}$.

Definition 16 (Types for realizers). *For each arithmetical formula* A *we define a type* $[A]$ *of* T *by induction on* A*:* $[P(t_1, \dots, t_n)] = \mathtt{S}$, $[A \wedge B] = [A] \times [B]$, $[A \vee B] = \mathtt{Bool} \times ([A] \times [B])$, $[A \to B] = [A] \to [B]$, $[\forall x A] = \mathtt{N} \to [A]$, $[\exists x A] = \mathtt{N} \times [A]$

Let now $p_0 := \pi_0$, $p_1 := \pi_0\pi_1$ and $p_2 := \pi_1\pi_1$. We define the notion of learning based realizability, which is relativized to a state $\sigma : \mathtt{N} \to \mathtt{N}$, and differs from Kreisel modified realizability for a single detail: if we realize an atomic formula, the atomic formula does not need to be true, unless the realizer is equal to the empty set when approximated at state σ.

Definition 17 (Learning-Based Realizability for $\mathsf{HA} + \mathsf{EM}_1$). *Assume* $\sigma : \mathtt{N} \to \mathtt{N}$ *is a closed term of Gödel's* T, t *is a closed term of* $\mathsf{T}_{\mathit{Class}}$, $C \in \mathcal{L}$ *is a closed formula, and* $t : [C]$. *Let* $\vec{t} = t_1, \dots, t_n : \mathtt{N}$. *We define the relation* $t \Vdash_\sigma C$ *by induction and by cases according to the form of* C*:*

1. $t \Vdash_\sigma P(\vec{t})$ *if and only if*

    (a) $\lambda f^{\mathtt{N}\to\mathtt{N}}\ t[f]$ *is a* EM_1*-update procedure*

    (b) $t[\sigma] = |\varnothing|$ *implies* $P(\vec{t}) = \mathtt{True}$

2. $t \Vdash_\sigma A \wedge B$ *if and only if* $\pi_0 t \Vdash_\sigma A$ *and* $\pi_1 t \Vdash_\sigma B$

3. $t \Vdash_\sigma A \vee B$ *if and only if either* $p_0 t[\sigma] = \mathtt{True}$ *and* $p_1 t \Vdash_\sigma A$*, or* $p_0 t[\sigma] = \mathtt{False}$ *and* $p_2 t \Vdash_\sigma B$

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

5. $t \Vdash_\sigma \forall x A$ *if and only if for all numerals* n*,* $tn \Vdash_\sigma A[n/x]$

6. $t \Vdash_\sigma \exists x A$ *if and only for some numeral* n*,* $\pi_0 t[\sigma] = n$ *in and* $\pi_1 t \Vdash_\sigma A[n/x]$

We define $t \Vdash A$ *if and only if* $t \Vdash_\sigma A$ *for all states* σ *of system* T*.*

Realizers are always computed with respect to a particular state of knowledge and hence they provide only approximated witnesses for the formulas they realize: instead of being certainties, witnesses are only *predictions*, based on the hope that the state is a good approximation of Φ. The most significant clause is the one for atomic formulas: the assertion $t \Vdash_\sigma P(\vec{t})$ has the following intuitive meaning. The approximation σ of Φ has led our realizability relation to predict that $P(\vec{t})$ is true. However, σ may be a bad approximation of Φ and $P(\vec{t})$ may be false. Providentially, $\lambda f^{\mathtt{N}\to\mathtt{N}}\ t[f]$ is a EM_1-update procedure and hence is able to correct some wrong value of σ with a right value: $t[\sigma]$ must be equal to $|(n, m)|$ for some m, n and thus σ must be corrected as to output m on input n. Thus, from any failure a realizer can always learn a new positive fact about Kleene's predicate T. This property is quite remarkable: we are really considering a model of computation that defines *self-correcting* programs, which are able to automatically repair themselves when they fail. Interestingly, this new ability of classical realizers is entirely a gift of classical logic when is applied on top of intuitionistic. We also argued in [2], that there is a close analogy between the way our realizers work and the modern scientific endeavor as described by Popper in his falsifiability theory. An approximation of the non computable function Φ is interpreted as a set of *hypotheses* on the true values of Φ; realizers infer predictions from these hypotheses; then these predictions are tested in real computations and if they are falsified, some of the current hypotheses about Φ are falsified too; finally, realizers correct some of the false hypotheses and build a better approximation.

We study as an instructive example the realizer of EM_1.

Example 1 (Realizer for EM_1). Define

$$\mathsf{Add} := \lambda x^{\mathtt{N}} \lambda y^{\mathtt{N}} \lambda z^{\mathtt{N}} \mathsf{if}\ \neg Txy\Phi|(x,y)| \wedge Txyz\ \mathsf{then}\ |(|(x,y)|, z)|\ \mathsf{else}\ |\varnothing|$$

and

$$\mathsf{E} := \lambda x^{\mathtt{N}} \lambda y^{\mathtt{N}}\ \langle Txy(\Phi|(x,y)|), \langle \Phi|(x,y)|,\ |\varnothing| \rangle,\ \mathsf{Add}xyz \rangle$$

We want to prove that $\mathsf{E} \Vdash \mathsf{EM}_1$. Let m, n be two numerals and let $\sigma : \mathtt{N} \to \mathtt{N}$ be a closed term of T. By definition

$$\mathsf{E}nm = \langle Txy(\Phi|(x,y)|), \langle \Phi|(x,y)|,\ |\varnothing|\rangle,\ \mathsf{Add}xyz\rangle$$

We have to prove that

$$\mathsf{E}nm \Vdash_\sigma \exists z^{\mathtt{N}}\ Tnmz \vee \forall z^{\mathtt{N}} \neg Txyz$$

We have that

$$p_0 \mathsf{E}nm[\sigma] = Tnm(\sigma|(n,m)|)$$

There are two cases:

1. $Tnm(\sigma|(n,m)|) = \mathtt{True}$. We have to prove that

$$p_1 \mathsf{E}nm \Vdash_\sigma \exists z^{\mathtt{N}}\ Tnmz$$

   Since for some numeral l

$$\pi_0 p_1 \mathsf{E}nm[\sigma] = \sigma|(n,m)| = l$$

   we have to prove that

$$\pi_1 p_1 \mathsf{E}m \Vdash_\sigma Tnml$$

   which is straightforward because $Tnml = \mathtt{True}$ by hypothesis and $\pi_1 p_1 E_P m = |\varnothing|$, so trivially $\lambda f^{\mathtt{N}\to\mathtt{N}}|\varnothing|$ is a EM_1-update procedure.

2. $Tnm(\sigma|(n,m)|) = \mathtt{False}$. We have to prove that

$$p_2 \mathsf{E}nm = \mathsf{Add}nm\ \Vdash_\sigma \forall z^{\mathtt{N}} \neg Tnmz$$

   i.e that, given any numeral l,

$$\mathsf{Add}nml \Vdash_\sigma \neg Tnml$$

   By the definition of realizer in this case, we have first to prove that $\lambda f^{\mathtt{N}\to\mathtt{N}} \mathsf{Add}nml[f]$ is a EM_1-update procedure, which is seen to be true just by definition of Add. Secondly, assume that $\mathsf{Add}nml[\sigma] = |\varnothing|$: we have to prove that $\neg Tnml = \mathtt{True}$, which is again immediate, since

$$\mathsf{Add}nml[\sigma] = \mathsf{if}\ \neg Tnm\sigma|(n,m)| \wedge Tnml\ \mathsf{then}\ |(|(n,m)|, l)|\ \mathsf{else}\ |\varnothing|$$

   and by hypothesis $Tnm(\sigma|(n,m)|) = \mathtt{False}$

We thus have concluded the proof. The behaviour of E in the proof can be explained as follows. σ is used by $\mathsf{E}nm$ as an approximation of Φ, i.e. to predict whether there exists a l such that $Tnml = \mathtt{True}$. In case (1), indeed such an l is exhibited by σ and everything goes well. But in case (2), σ does not yield such an l: E hopes that σ is nevertheless a good approximation of the Skolem function Φ and so declares that $\forall z^{\mathtt{N}} \neg Txyz$. But a counterexample could anyway be encountered, so for every given l, $\mathsf{Add}nml$ tests whether $Tnml = \mathtt{True}$ and in this case returns an update: from a failure a non trival witness is learned. This behaviour is well explained in term of 1-Backtracking games (see [3], [1]) Any classical proof using only excluded middle over semi-decidable formulas may be interpreted by some learning based realizer.

Theorem 11 (Soundness of Realizability for $\mathsf{HA} + \mathsf{EM}_1$). *If A is a closed formula provable in $\mathsf{HA} + \mathsf{EM}_1$ (see [2]), then there exists $t \in \mathsf{T}_{Class}$ such that $t \Vdash A$.*

Proof. By Aschieri and Berardi [2].

As anticipated in the introduction, from learning based realizers we can extract algorithms of system T.

Theorem 12 (Program Extraction via Learning Based Realizability). *Let t be a term of T_{Class} and suppose that $t \Vdash \forall x^{\mathtt{N}} \exists y^{\mathtt{N}} Pxy$, with P atomic. Then, from t one can effectively define a term u of Gödel's system T such that for every numeral n, $Pn(un) = \mathtt{True}$.*

Proof. Let
$$v := \lambda m^{\mathtt{N}} \lambda f^{\mathtt{N} \rightarrow \mathtt{N}}\ \pi_1(tm)[f]$$
v is of type $\mathtt{N} \rightarrow (\mathtt{N} \rightarrow \mathtt{N}) \rightarrow \mathtt{N}$. By definition of realizability, for every σ and numeral n
$$\pi_1(tn) \Vdash_{\sigma} Pn(\pi_0(tn)[\sigma])$$
and hence for every numeral n, $\lambda f^{\mathtt{N} \rightarrow \mathtt{N}}\ \pi_1(tn)[f] = v_n$ is a typed EM_1-update procedure.

By theorem 10, there exists a term $\mathsf{zero} : \mathtt{N} \rightarrow (\mathtt{N} \rightarrow \mathtt{N})$ of T such that $v_n(\mathsf{zero}_n) = |\varnothing|$ for every numeral n. By construction, zero_n is also a state of knowledge. Define
$$u := \lambda m^{\mathtt{N}} \pi_0(tm)[\mathsf{zero}_m]$$
and fix a numeral n. By unfolding the definition of realizability with respect to the state zero_n, we have that
$$tn \Vdash_{\mathsf{zero}_n} \exists y^{\mathtt{N}} Pny$$
and hence
$$\pi_1(tn) \Vdash_{\mathsf{zero}_n}\ Pn(un)$$
that is to say
$$v_n(\mathsf{zero}_n) = |\varnothing| \implies Pn(un) = \mathtt{True}$$
and therefore
$$Pn(un) = \mathtt{True}$$
which is the thesis.

We are now able to prove a version of the classic theorem of Gödel, characterizing the class of functions provably total in PA as the class of functions representable in system T.

Theorem 13 (Provably Total Functions of PA). *If $\mathsf{PA} \vdash \forall x^{\mathtt{N}} \exists y^{\mathtt{N}} Pxy$, then there exists a term u of Gödel's system T such that for every numeral n, $Pn(un) = \mathtt{True}$.*

PROOF. As usual, it is not restrictive to assume that $Pxy = Tnxy$ for some numeral n. Starting from the assumption that

$$\mathsf{PA} \vdash \forall x^{\mathtt{N}} \exists y^{\mathtt{N}} Tnxy$$

by Gödel or Kolmogorov double negation translation (see for instance [19]), we have that

$$\mathsf{HA} \vdash \forall x^{\mathtt{N}} \neg\neg \exists y^{\mathtt{N}} Tnxy$$

Therefore

$$\mathsf{HA} + \mathsf{EM}_1 \vdash \forall x^{\mathtt{N}} \exists y^{\mathtt{N}} Tnxy$$

and so there is a term t of $\mathsf{T}_{\text{Class}}$ such that

$$t \Vdash \forall x^{\mathtt{N}} \exists y^{\mathtt{N}} Tnxy$$

By theorem 12, there exists a term u of Gödel's system T such that for all numerals n

$$Pnu(n) = \mathtt{True}$$

Remark 1. Theorem 13 is usually derived through a combination of Gödel's negative translation followed by either Friedman's translation and modified realizability or Dialectica translation (see for example Kohlenbach [14]). In our approach, we instead perform only the first translation and we need no additional ones, because we can directly interpret the resulting proof as it is. The real advantages, however, arise when interpreting proofs in $\mathsf{HA} + \mathsf{EM}_1$, because no translation whatsoever is made and classical reasoning is directly analyzed (with of course great benefit for the actual understanding of the extracted program). Moreover, by extending learning based realizability to all PA, it will be even possible eliminate the first translation.

6.4. Zeros for k-ary Update Procedures

Thanks to the adequacy theorem, we are also able to give a new constructive proof of Avigad's theorem for k-ary update procedures. Again the definition is slightly different from Avigad's, but analogous.

Intuitively, a k-ary update procedure, with $k \geq 2$, is a functional which takes as input a finite sequence $f = f_1, \ldots, f_k$ of functions approximating some oracles, such that each one of those functions is defined in terms of the previous ones. Then, it uses those functions to compute some witnesses for some provable Σ_1^0 formula of PA. Afterwards, it checks whether the result of its computation is sound. If it is not, it identifies some wrong value $f_i(n)$ used in the computation and it corrects it with a new one.

Definition 18 (Typed Update Procedures). *Fix a primitive recursive bijective coding $\|\_\| : (\mathbb{N}^3 \cup \{\varnothing\}) \rightarrow \mathbb{N}$ of $\varnothing$ and of triples of natural numbers into natural numbers. A k-ary* typed update procedure *$k \in \mathbb{N}$, $k \geq 2$ is a term $\mathcal{U} : (\mathtt{N} \rightarrow \mathtt{N})^k \rightarrow \mathtt{N}$ of Gödel's T such that the following holds:*

1. *for all sequences $f = f_1, \ldots, f_k$ of closed type-$\mathtt{N} \rightarrow \mathtt{N}$ terms of T, $\mathcal{U}f = \|(i, n, m)\| \implies 1 \leq i \leq k$.*

2. *for all sequences* $f = f_1, \dots, f_k$ *and* $g = g_1, \dots, g_k$ *of closed type-*$\mathtt{N} \to \mathtt{N}$ *terms of* T *and for all* $1 \le i < k$*, if*

   i) for all $j < i$*,* $f_j = g_j$*;*

   ii) $\mathcal{U}f = \|(i, n, m)\|$*,* $g_i(n) = m$ *and* $\mathcal{U}g = \|(i, h, l)\|$

   then $h \ne n$*.*

If $\mathcal{U}$ *is a* k*-ary typed update procedure, a* zero *for* $\mathcal{U}$ *is a sequence* $f = f_1, \dots, f_k$ *of closed type-*$\mathtt{N} \to \mathtt{N}$ *terms of* T *such that* $\mathcal{U}f = \|\varnothing\|$*.*

Condition ii) of definition 18 is explained in the same way as with unary update procedures, but it requires condition i) to make sense: the values of the i-th function depend on the values of some of the functions f_j, with $j < i$, and learning on level i is possible only if all the lower levels j have "stabilized" (see Avigad [4]).

We need a stronger version of theorem 10.

Theorem 14 (Second Zero Theorem for Unary Typed Update Procedures). *Let* $\mathcal{U} : (\mathtt{N} \to \mathtt{N})^{k+1} \to \mathtt{N}$ *be a term of* T *such that for all closed type-*$\mathtt{N} \to \mathtt{N}$ *terms* $f_1, \dots, f_k$ *of* T*,* $\mathcal{U}f_1 \dots f_k$ *is a typed unary update procedure. Then one can constructively define a closed term* $\varepsilon : (\mathtt{N} \to \mathtt{N})^k \to (\mathtt{N} \to \mathtt{N})$ *of* T *such that for all closed type-*$\mathtt{N} \to \mathtt{N}$ *terms* $f_1, \dots, f_k$ *of* T

$$\mathcal{U}f_1 \dots f_k(\varepsilon f_1 \dots f_k) = |\varnothing|$$

Proof. First, for any term $h : \mathtt{N} \to \mathtt{N}$, define

$$\mathcal{L}^h := \lambda\langle \mathcal{M}, g\rangle^{\mathsf{M}_\mathtt{N}} \langle \mathcal{M}, \lambda n^{\mathtt{N}} h(g(n))\rangle$$

The same proof of proposition 3 (in the case of constants of type $\mathtt{N} \to \mathtt{N}$) shows that for all closed terms h of T and $s \in$w.i., $\mathcal{L}^h$ gmc_s h. Define

$$\mathcal{N}_s := \lambda h_1^{\mathtt{N}\to\mathtt{N}} \dots \lambda h_k^{\mathtt{N}\to\mathtt{N}} [\![\mathcal{U}]\!]_s \mathcal{L}^{h_1} \dots \mathcal{L}^{h_k} [\![\Phi]\!]_s$$

and fix closed type-$\mathtt{N} \to \mathtt{N}$ terms $f_1, \dots, f_k$ of T. By the adeguacy theorem 8, for all $s \in$w.i., $[\![\mathcal{U}]\!]_s$ gmc_s $\mathcal{U}$ and hence

$$\mathcal{M}_s := \mathcal{N}_s f_1 \dots f_k \ \mathsf{gmc}_s \ \mathcal{U}f_1 \dots f_k \Phi$$

So, for all $s \in$ w.i., $\pi_0(\mathcal{M}_s)$ is a modulus of convergence for $\lambda m^{\mathtt{N}} \mathcal{U}f_1 \dots f_k(s_m)$. Define by recursion a term s such that $s_0 := 0^{\mathtt{N}\to\mathtt{N}}$ and $s_{n+1} := s_n \oplus \mathcal{U}f_1 \dots f_k s_n$. Then $s \in$w.i. by proposition 4, since $\mathcal{U}f_1 \dots f_k$ is a typed unary update procedure. So $\pi_0(\mathcal{M}_s)$ is a modulus of convergence for $\lambda m^{\mathtt{N}}\ \mathcal{U}f_1 \dots f_k(s_m)$. If we choose $h := \lambda m^{\mathtt{N}} m + 1$ and set $j := \pi_0(\mathcal{M}_s)h$, we have that

$$\mathcal{U}f_1 \dots f_k(s_j) = \mathcal{U}f_1 \dots f_k(s_{j+1})$$

by definition of modulus of convergence. So let

$$\varepsilon f_1 \dots f_k := s_{(\pi_0(\mathcal{M}_s)h)+1}$$

Then

$$\begin{aligned} s_{j+2} &= s_{j+1} \oplus \mathcal{U} f_1 \dots f_k(s_{j+1}) \\ &= (s_j \oplus \mathcal{U} f_1 \dots f_k(s_j)) \oplus \mathcal{U} f_1 \dots f_k(s_{j+1}) \\ &= (s_j \oplus \mathcal{U} f_1 \dots f_k(s_j)) \oplus \mathcal{U} f_1 \dots f_k(s_j) \\ &= s_j \oplus \mathcal{U} f_1 \dots f_k(s_j) \\ &= s_{j+1} \end{aligned}$$

and hence it must be that

$$\mathcal{U} f_1 \dots f_k(\varepsilon f_1 \dots f_k) = \mathcal{U} f_1 \dots f_k(s_{j+1}) = |\varnothing|$$

which is the thesis.

We are now able to prove the zero theorem for n-ary typed update procedures, following the idea of Avigad's original construction.

Theorem 15 (Zero Theorem for k-ary Typed Update Procedures). *Let*

$$\mathcal{U} : (\mathtt{N} \to \mathtt{N})^k \to \mathtt{N}$$

be a k-ary typed update procedure. Then one can constructively define terms $\varepsilon_1, \dots, \varepsilon_k$ of Gödel's T such that

$$\mathcal{U} \varepsilon_1 \dots \varepsilon_k = |\varnothing|$$

Proof. By induction on k. The case $k = 1$ has been treated in theorem 10. Therefore, suppose $k \geq 2$. Define

$$\mathcal{U}_k := \lambda g_1^{\mathtt{N}\to\mathtt{N}} \dots \lambda g_{k-1}^{\mathtt{N}\to\mathtt{N}} \text{ if } \mathcal{U} g_1 \dots g_{k-1} = \|(k, n, m)\| \text{ then } |(n, m)| \text{ else } |\varnothing|$$

Since for all closed type-$\mathtt{N} \to \mathtt{N}$ terms $f_1, \dots, f_{k-1}$ of T, $\mathcal{U}_k f_1 \dots f_{k-1}$ is a typed unary update procedure, by theorem 14 we can constructively define a term ε_k of T such that for all closed type-$\mathtt{N} \to \mathtt{N}$ terms $f_1, \dots, f_{k-1}$ of T

$$\mathcal{U}_k f_1 \dots f_{k-1}(\varepsilon_k f_1 \dots f_{k-1}) = |\varnothing|$$

and hence for all n, m

$$\mathcal{U} f_1 \dots f_{k-1}(\varepsilon_k f_1 \dots f_{k-1}) \neq \|(k, n, m)\|$$

This implies that

$$\lambda g_1^{\mathtt{N}\to\mathtt{N}} \dots \lambda g_{k-1}^{\mathtt{N}\to\mathtt{N}}\, \mathcal{U} g_1 \dots g_{k-1}(\varepsilon g_1 \dots g_{k-1})$$

is a typed $k-1$-ary update procedure (if $k = 2$ one, trivially, needs also a projection and hence to consider $\mathcal{U}_1$ instead $\mathcal{U}$). By induction hypothesis, we can constructively define terms $\varepsilon_1, \dots, \varepsilon_{k-1}$ of Gödel's T such that

$$\mathcal{U} \varepsilon_1 \dots \varepsilon_{k-1}(\varepsilon_k \varepsilon_1 \dots \varepsilon_{k-1}) = |\varnothing|$$

which is the thesis.

An important corollary of theorem 15, is the termination of the epsilon substitution method for first order Peano Arithmetic.

Theorem 16 (Termination of Epsilon Substitution Method for PA). *The H-process (as defined in Mints [16]) of the epsilon substitution method for* PA *always terminates.*

PROOF. See Avigad [4]

[1] F. Aschieri, *Learning, Realizability and Games in Classical Arithmetic*, PhD Thesis, 2011. `http://arxiv.org/abs/1012.4992`

[2] F. Aschieri, S. Berardi, *Interactive Learning-Based Realizability for Heyting Arithmetic with* EM_1, Logical Methods in Computer Science, vol. 6 issue 3, 2010

[3] F. Aschieri, *Interactive Learning Based Realizability and 1-Backtracking Games*, Proceedings of Classical Logic and Computation, Electronic Proceedings in Theoretical Computer Science, vol. 47, 2011.

[4] Jeremy Avigad: *Update Procedures and the 1-Consistency of Arithmetic*. Math. Log. Q. 48(1): 3-13 (2002).

[5] S. Berardi, U. de' Liguoro, *Interactive Realizers and Monads*

[6] S. Berardi, *Personal Communication*

[7] S. Berardi, *Classical Logic as Limit Completion*, MSCS, Vol. 15, n.1, 2005, pp.167-200.

[8] S. Berardi, T. Coquand, S. Hayashi, *Games with 1-Bactracking*, APAL 2010, to appear.

[9] S. Berardi, U. de' Liguoro, *A calculus of realizers for* EM_1*-Arithmetic*, Proceedings of Computer Science Logic 2008, in LNCS 5213, pag 215-229 (2008)

[10] U. Berger, *Continuous Semantics for Strong Normalization*, Lecture Notes in Computer Science 3526, 23–34, 2005

[11] J.-Y. Girard, *Proofs and Types*, Cambridge University Press (1989)

[12] Goldblatt, *Lectures on the Hyperreals*, Springer-Verlag, New York, 1998

[13] S. C. Kleene, *On the Interpretation of Intuitionistic Number Theory*, Journal of Symbolic Logic 10(4), pag 109-124 (1945)

[14] U. Kohlenbach, *Applied Proof Theory*, Springer-Verlag, Berlin, Heidelberg, 2008

[15] G. Kreisel, *Interpretation of analysis by means of constructive functionals of nite types*, Heyting, A. (ed.), Constructivity in Mathematics, pp. 101128. North- Holland, Amsterdam (1959).

[16] G. Mints, S. Tupailo, W. Bucholz, *Epsilon Substitution Method for Elementary Analysis*, Archive for Mathematical Logic, volume 35, 1996

[17] G. Mints, *Epsilon Substitution Method for Predicate Logic*, Proceedings of Classical Logic and Computation 2010, Electronic Proceedings in Theoretical Computer Science, vol. 47, 2011.

[18] H. Schwichtenberg, *On Bar Recursion of Type 0 and 1*, The Journal of Symbolic Logic, volume 44, number 3, 1979

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

[20] T. Skolem, *Selected Works in Logic* Universitets for laget, Oslo, 1970. Edited by Fenstad, J. E.

[21] A.S. Troelstra, editor. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, Springer Lecture Notes in Mathematics 344, 1973.