An interpretation of the Sigma-2 fragment of classical Analysis in System T

Danko Ilik

▶ To cite this version:

HAL Id: hal-01092411

https://hal.inria.fr/hal-01092411

AN INTERPRETATION OF THE SIGMA-2 FRAGMENT
OF CLASSICAL ANALYSIS IN SYSTEM T

DANKO ILIK

ABSTRACT. We show that it is possible to define a realizability interpretation for the Σ_2-fragment of classical Analysis using Gödel's System T only. This supplements a previous result of Schwichtenberg regarding bar recursion at types 0 and 1 by showing how to avoid using bar recursion altogether. Our result is proved via a conservative extension of System T with an operator for composable continuations from the theory of programming languages due to Danvy and Filinski. The fragment of Analysis is therefore essentially constructive, even in presence of the full Axiom of Choice schema: Weak Church's Rule holds of it in spite of the fact that it is strong enough to refute the formal arithmetical version of Church's Thesis.

1. INTRODUCTION

In the middle of the 20th century, Kurt Gödel showed how to give a computational interpretation and a relative consistency proof of intuitionistic Arithmetic via his System T of equations between functionals definable by primitive recursion (in higher types) [11, 10]. Thanks to the fact that the induction axiom intuitionistically proves its own double negation translation, the interpretation also applies to classical Arithmetic. However, since the Axiom of Choice,

$$\text{(AC)} \qquad \forall x \exists y A(x, y) \rightarrow \exists f \forall x A(x, f(x)),$$

does not intuitionistically prove its double negation translation, the interpretation does not apply to classical Analysis. Gödel was of course aware of this fact and suggested [21, §2.43] that an extension of System T is needed which can interpret the logical schema,

$$\neg\neg\forall x(A(x) \vee \neg A(x)),$$

known as Kuroda's Conjecture [24], intuitionistically equivalent to the nowadays better known schema of Double Negation Shift,

$$\text{(DNS)} \qquad \forall x \neg\neg B(x) \rightarrow \neg\neg\forall x B(x).$$

Gödel must have also been aware of the difficulty involved in giving a computational interpretation to Kuroda's Conjecture, for already his 1941 lecture at Yale [11] considers the special case when the formula $A(x)$ is $\exists y\, T(x, x, y)$ — where T is Kleene's predicate verifying that the Turing machine with code x, when run on input x, terminates with code y — which directly proves (see [33]) the *negation* of the formal arithmetical version of Church's Thesis,

$$\text{(CT}_0\text{)} \qquad \forall x^{\mathbb{N}} \exists y^{\mathbb{N}} A(x, y) \rightarrow \exists e^{\mathbb{N}} \forall x^{\mathbb{N}} \exists u^{\mathbb{N}}(T(e, x, u) \wedge A(x, U(u))).$$

In spite of that, Spector and Kreisel [31, 9] managed to give a computational interpretation of DNS by extending System T with bar recursion, the computational adequacy of which was shown using a formal version of Brouwer's principle of

Bar Induction [31, 14]. This approach to extracting computational content from proofs in Analysis via an extension of the primitive recursive System T with a *general recursive* schema, has been much refined over the 50 years since it appeared [22, 32, 20, 2, 1, 3, 4, 29] and has been applied to obtain results in Analysis proper, notably in Kohlenbach's Proof Mining programme [20].

Nevertheless, as Schwichtenberg showed already in 1979 [27], higher type primitive recursion is closed over the schema of bar recursion at types 0 and 1, and since a previous analysis of Kreisel [31, §12.2] shows that those low types are sufficient for interpreting the classical Axiom of Choice for formulas of the form $\exists \alpha \in \mathbb{N} \to \mathbb{N} \forall x \in \mathbb{N} A_0(\alpha, x)$, A_0-quantifier-free, we in fact known that we should need no more than the primitive recursive functionals themselves in order to give a realizability interpretation of the uniformly realizable part of Analysis. Yet, it has remained unclear up to this day how to avoid using general recursive schemata altogether.

One alternative is offered by the use of so called computational side-effects (control operators) from the theory of programming languages. Krivine [23] used a realizability interpretation based on a virtual machine which can execute lambda calculus terms, extended with a control operator and a special machine instruction called "quote". Herbelin [13] gave a more direct approach based on rewrite rules for a type theory extended with a control operator and a coinductive treatment of the existential quantifier. Both approaches rely on a proper extension of System T that can give a computational interpretation to full classical logic. However, given that there are classically true arithmetic statements that do not have a recursive realizer, it is not clear what the meaning of control operators outside the Σ_1^0-fragment is.

In this paper, we show that computational side-effects are not needed in the language of realizers, that is, although they are conceptually essential, control operators can be seen as a *meta-mathematical* technique. Proofs of the Σ_2-fragment of Analysis, with the full Axiom of Choice, are essentially constructive and realizable by System T terms only. The soundness of the interpretation relies on a form of Markov's Principle (the SHIFT rule), rather than full classical logic [32, 23, 13] or continuity principles [31, 14, 1, 3].

2. CONSERVATIVE EXTENSION OF SYSTEM T WITH OPERATORS FOR COMPOSABLE CONTINUATIONS

The constructive interpretation of proofs of Section 3 is based on Gödel's System T in its lambda-calculus formulation [33, 28]. Nevertheless, we consider it a conceptual advantage to use an intermediate system, the System T^+ obtained when System T is extended with a control operator for composable continuations, the so called call-by-name variant of the *shift* operator of Danvy and Filinski [7]. The control operator is a key conceptual ingredient that led us to the interpretation [12, 15, 16, 19].

The goal of this technical section is to prove the following conservativity theorem, as well as prove that suitable equations important for Section 3 hold of System $T^{(+)}$ (Proposition 1).

Normalization Theorem. *Every term of System T^+ maps to a term in normal form of System T.*

The *types* of System $T^{(+)}$ are $\mathbb{N}$, functions $\mathbb{N} \to \mathbb{N}$, functionals (ex. $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$), and Cartesian products of these. Types will be denoted by σ, τ:

$$\mathcal{T} \ni \sigma, \tau ::= \mathbb{N} \mid \sigma \to \tau \mid \sigma * \tau.$$

Terms of System T^+ are constants associated with a sequent $\sigma_1; \ldots; \sigma_n \vdash \tau$ (also written $\gamma \vdash \tau$ for γ a finite ordered list of types) which means that a term is of type τ and the free variables that appear in the term are of types $\sigma_1, \ldots, \sigma_n$. The terms, marked in sans-serif face, are defined inductively as follows.

$$\text{hyp } \frac{}{(\sigma; \gamma) \vdash \sigma} \qquad \text{wkn } \frac{\gamma \vdash \sigma}{(\tau; \gamma) \vdash \sigma} \qquad \text{lam } \frac{(\sigma; \gamma) \vdash \tau}{\gamma \vdash \sigma \to \tau}$$

$$\text{app } \frac{\gamma \vdash \sigma \to \tau \quad \gamma \vdash \sigma}{\gamma \vdash \tau} \qquad \text{pair } \frac{\gamma \vdash \sigma \quad \gamma \vdash \tau}{\gamma \vdash \sigma * \tau} \qquad \text{fst } \frac{\gamma \vdash \sigma * \tau}{\gamma \vdash \sigma}$$

$$\text{snd } \frac{\gamma \vdash \sigma * \tau}{\gamma \vdash \tau} \qquad \text{zero } \frac{}{\gamma \vdash \mathbb{N}} \qquad \text{succ } \frac{\gamma \vdash \mathbb{N}}{\gamma \vdash \mathbb{N}}$$

$$\text{rec } \frac{\gamma \vdash \mathbb{N} \quad \gamma \vdash \sigma \quad \gamma \vdash \mathbb{N} \to \sigma \to \sigma}{\gamma \vdash \sigma} \qquad \text{shift } \frac{(\mathbb{N} \to \sigma \to \mathbb{N}; \gamma) \vdash \mathbb{N}}{\gamma \vdash \sigma}$$

For example, the Ackermann function of type $\mathbb{N} \to \mathbb{N} \to \mathbb{N}$ in lambda calculus notation,

$$A := \lambda m. \, R \, m(\lambda n.n + 1)(\lambda m'.\lambda u.\lambda n. \, R \, n(u1)(\lambda n'.\lambda w.uw)),$$

where R is a constant such that

$$R \, 0ab = a$$
$$R(n + 1)ab = bn(R \, nab),$$

is defined using the following term:

```
lam
   (rec hyp(lam(succ hyp))
      (lam
         (lam
            (lam
               (rec hyp(app(wkn hyp)(succ zero))
                  (lam(lam(app(wkn(wkn(wkn hyp))) hyp)))))))).
```

This style of presenting lambda calculus terms, known as deBruijn convention, keeps the language of terms first-order, that is, it avoids problems related to handling variables as names that we have to keep track of externally to the system: hyp denotes the variable corresponding to the nearest preceding lambda abstraction lam or a shift, wkn hyp denotes the second most recent introduced variable, wkn wkn hyp the third one, and so on. Seen as natural numbers (hyp is 0, wkn is successor), these are just so called deBruijn indices.

The terms of System T have a great computational potential, they can compute any higher-type primitive recursive function. There is, however, a subclass of terms

which denote computations that have finished, the so called terms *in normal form*. These will be sufficient for denoting programs and data extracted from proofs in classical Analysis in Section 3. Technically, the normal forms are known as β-normal in reference to the associated β-reduction relation. Here we give a direct inductive characterization of normal terms ($\gamma \Vdash_{\mathsf{r}} $),

$$\mathsf{e}\ \frac{\gamma \Vdash_{\mathsf{e}} \sigma}{\gamma \Vdash_{\mathsf{r}} \sigma} \qquad \mathsf{lam}\ \frac{(\sigma;\gamma) \Vdash_{\mathsf{r}} \tau}{\gamma \Vdash_{\mathsf{r}} \sigma \to \tau} \qquad \mathsf{pair}\ \frac{\gamma \Vdash_{\mathsf{r}} \sigma \quad \gamma \Vdash_{\mathsf{r}} \tau}{\gamma \Vdash_{\mathsf{r}} \sigma * \tau}$$

$$\mathsf{zero}\ \frac{}{\gamma \Vdash_{\mathsf{r}} \mathbb{N}} \qquad \mathsf{succ}\ \frac{\gamma \Vdash_{\mathsf{r}} \mathbb{N}}{\gamma \Vdash_{\mathsf{r}} \mathbb{N}}$$

defined at the same time with the so called *neutral* terms ($\gamma \Vdash_{\mathsf{e}}$),

$$\mathsf{hyp}\ \frac{}{(\sigma;\gamma) \Vdash_{\mathsf{e}} \sigma} \qquad \mathsf{wkn}\ \frac{\gamma \Vdash_{\mathsf{r}} \sigma}{(\tau;\gamma) \Vdash_{\mathsf{e}} \sigma} \qquad \mathsf{app}\ \frac{\gamma \Vdash_{\mathsf{e}} \sigma \to \tau \quad \gamma \Vdash_{\mathsf{r}} \sigma}{\gamma \Vdash_{\mathsf{e}} \tau}$$

$$\mathsf{fst}\ \frac{\gamma \Vdash_{\mathsf{e}} \sigma * \tau}{\gamma \Vdash_{\mathsf{e}} \sigma} \qquad \mathsf{snd}\ \frac{\gamma \Vdash_{\mathsf{e}} \sigma * \tau}{\gamma \Vdash_{\mathsf{e}} \tau} \qquad \mathsf{rec}\ \frac{\gamma \Vdash_{\mathsf{e}} \mathbb{N} \quad \gamma \Vdash_{\mathsf{r}} \sigma \quad \gamma \Vdash_{\mathsf{r}} \mathbb{N} \to \sigma \to \sigma}{\gamma \Vdash_{\mathsf{e}} \sigma}.$$

Neutral terms correspond to computation that are "blocked": neutral terms are those that contain open/free variables that block a β-reduction step from happening.

A property that will be later used (proof of Corollary 1) follows directly from the shape of normal forms: any closed normal term of type $\mathbb{N}$, i.e. a term of type $\emptyset \Vdash_{\mathsf{r}} \mathbb{N}$, is actually a numeral, that is, built only from zero and succ-terms. This follows because *closed* normal terms cannot be neutral: a neutral term necessarily has at least one free variable.

We are now ready to state our first theorem precisely.

Normalization Theorem. *There is a normalization procedure $\downarrow \llbracket - \rrbracket$ such that, for every term p of System T^+ of type $\gamma \vdash \tau$, the term $\downarrow \llbracket p \rrbracket$ is a term in normal form of System T of the same type ($\gamma \Vdash_{\mathsf{r}} \tau$).*

This theorem is not a standard fact from the theory of lambda calculus with control operators. As a matter of fact, we present the first proof that control operators can be completely eliminated from System T.

Specialists will recognize the proof method as a *normalization-by-evaluation* [5] or *type-directed partial evaluation* [6] argument. Nevertheless, the addition of control operators requires us to perform the proof in so called *continuation-passing style*; a similar technique has been used to provide a constructive completeness proof for non-minimal intuitionistic logic when disjunction and the existential quantifier are present [15, 17].

The proof is constructive and can be formalized in a suitable predicative meta-theory such as Martin-Löf Type Theory (for a modern formulation see [34]). At her convenience, the reader may find this full mechanization of the proof in Agda notation in [18]; one can also use this machine-checked constructive proof directly in order to compute, for example, that the term for the Ackermann function A(3,2) really evaluates to $\underbrace{\mathsf{succ} \cdots \mathsf{succ}}_{29 \text{ times}} \mathsf{zero}$.

Proof of Normalization Theorem. Our goal is to define an *evaluation* function,

$$\llbracket (-) \rrbracket : \gamma \vdash \sigma \Rightarrow \gamma \Vdash \sigma$$

that maps a term p of type $\gamma \vdash \sigma$ to a *forcing*[1] set $\gamma \Vdash \sigma$, together with a *reification* function,

$$\downarrow (-) : \gamma \Vdash \sigma \Rightarrow \gamma \vdash_{\mathsf{T}} \sigma,$$

that extracts normal forms (without shift!) from the forcing set. Composing the two function, $p \mapsto\, \downarrow [\![p]\!]$, gives a normalization procedure. (The notation $a\, b\, c \mapsto d$ will be a compact form of $a \mapsto b \mapsto c \mapsto d$.)

We first need a precise way to speak about extensions of type contexts γ (imposed by the fact that we want to normalize potentially open terms). This is formalized by the initial segment, or prefix, preorder $\geq$ as follows.

$$\geq_{\mathrm{refl}} \frac{}{\gamma \geq \gamma} \qquad \geq_{\mathrm{cons}} \frac{\gamma_2 \geq \gamma_1}{(\sigma; \gamma_2) \geq \gamma_1}$$

For example, the proof of $\tau_1; \tau_2; \gamma \geq \gamma$ will be denoted by $\geq_{\mathrm{cons}}\geq_{\mathrm{cons}}\geq_{\mathrm{refl}}$. The transitivity of $\geq$ is proven as the (right-associative) operation $(-) \cdot (-)$, defined by recursion on the construction of the proofs of $\gamma_3 \geq \gamma_2$ and $\gamma_2 \geq \gamma_1$.

$$(-) \cdot (-) : \gamma_3 \geq \gamma_2 \Rightarrow \gamma_2 \geq \gamma_1 \Rightarrow \gamma_3 \geq \gamma_1$$
$$\geq_3 \cdot \geq_{\mathrm{refl}} =\, \geq_3$$
$$\geq_{\mathrm{refl}} \cdot \geq_2 =\, \geq_2$$
$$(\geq_{\mathrm{cons}}\geq_3)\cdot \geq_2 =\, \geq_{\mathrm{cons}} (\geq_3 \cdot \geq_2)$$

In this definition, and henceforth, the notation $\geq_n$ will be used to denote a (hypothetical) proof of $\gamma_n \geq \gamma_k$. For example, given a proof $\geq_2$ of $\gamma_2 \geq \sigma; \gamma_1$, one can prove $\gamma_2 \geq \gamma_1$ by the denotation $\geq_2 \cdot \geq_{\mathrm{cons}}\geq_{\mathrm{refl}}$.

We can now define precisely the *forcing* set $\gamma \Vdash \sigma$. This is done simultaneously with the *strong forcing* set $\gamma \Vdash_{\mathsf{S}} \sigma$, itself defined inductively following the construction of the type σ.

$$\gamma \Vdash \sigma = \forall \gamma_1 \geq \gamma\, (\forall \gamma_2 \geq \gamma_1\, (\gamma_2 \Vdash_{\mathsf{S}} \sigma \Rightarrow \gamma_2 \vdash_{\mathsf{T}} \mathbb{N}) \Rightarrow \gamma_1 \vdash_{\mathsf{T}} \mathbb{N})$$

$$\gamma \Vdash_{\mathsf{S}} \mathbb{N} = \gamma \vdash_{\mathsf{T}} \mathbb{N}$$
$$\gamma \Vdash_{\mathsf{S}} (\sigma \to \tau) = \forall \gamma' \geq \gamma(\gamma' \Vdash \sigma \Rightarrow \gamma' \Vdash \tau)$$
$$\gamma \Vdash_{\mathsf{S}} (\sigma * \tau) = \gamma \Vdash \sigma \times \gamma \Vdash \tau$$

We will also need $\gamma' \Vvdash \gamma$, the component-wise extension of the forcing relation defined as follows ([] denotes the empty context).

$$\gamma' \Vvdash [] = \top$$
$$\gamma' \Vvdash (\sigma; \gamma) = (\gamma' \Vdash \sigma) \times (\gamma' \Vvdash \gamma)$$

The first equation defines the forcing of the empty context to be the singleton set $\top$, whose unique inhabitant is denoted tt. The symbol $\times$ constructs a Cartesian product i.e. pair type in the ambient type theory; the components of a Cartesian product can be accessed by the projection operations proj_1 and proj_2.

---

[1]The terminology "forcing" comes from similarity of our construction with Kripke models which does not intentionally refer to the forcing from Set Theory.

The following operations (lemmas) show that $\Vdash_{\overline{r}}, \Vdash_{\overline{e}}, \Vdash, \Vdash_{\overline{s}}, \Vdash$ are all monotone with respect to the prefix preorder. The last two operations are defined by induction on types and contexts, respectively.

$$\ulcorner(-)\urcorner^{(-)} : \gamma_2 \geq \gamma_1 \Rightarrow \gamma_1 \Vdash_{\overline{r}} \sigma \Rightarrow \gamma_2 \Vdash_{\overline{r}} \sigma$$
$$\ulcorner H \urcorner^{\geq \text{refl}} = H$$
$$\ulcorner H \urcorner^{\geq_{\text{cons}} \geq_2} = \mathsf{e}(\mathsf{wkn}(\ulcorner H \urcorner^{\geq_2}))$$

$$\llcorner(-)\lrcorner_{(-)} : \gamma_2 \geq \gamma_1 \Rightarrow \gamma_1 \Vdash_{\overline{e}} \sigma \Rightarrow \gamma_2 \Vdash_{\overline{e}} \sigma$$
$$\llcorner H \lrcorner_{\geq \text{refl}} = H$$
$$\llcorner H \lrcorner_{\geq_{\text{cons}} \geq_2} = \mathsf{wkn}(\mathsf{e}(\llcorner H \lrcorner_{\geq_2}))$$

$$\lceil(-)\rceil^{(-)} : \gamma_2 \geq \gamma_1 \Rightarrow \gamma_1 \Vdash \sigma \Rightarrow \gamma_2 \Vdash \sigma$$
$$\lceil H \rceil^{\geq \text{refl}} = H$$
$$\lceil H \rceil^{\geq_{\text{cons}} \geq_2} = \geq_3 \mapsto H(\geq_3 \cdot \geq_{\text{cons}} \geq_2)$$

$$\lfloor(-)\rfloor^{\sigma}_{(-)} : \gamma_2 \geq \gamma_1 \Rightarrow \gamma_1 \Vdash_{\overline{s}} \sigma \Rightarrow \gamma_2 \Vdash_{\overline{s}} \sigma$$
$$\lfloor H \rfloor^{\sigma \to \tau}_{\geq_2} = \geq_3 \mapsto H(\geq_3 \cdot \geq_2)$$
$$\lfloor(H_1, H_2)\rfloor^{\sigma * \tau}_{\geq_2} = \lceil H_1 \rceil^{\geq_2}, \lceil H_2 \rceil^{\geq_2}$$
$$\lfloor H \rfloor^{\mathbb{N}}_{\geq_2} = \ulcorner H \urcorner^{\geq_2}$$

$$\llbracket(-)\rrbracket^{(-)}_{\gamma_2} : \gamma_2 \geq \gamma_1 \Rightarrow \gamma_1 \Vdash \gamma \Rightarrow \gamma_2 \Vdash \gamma$$
$$\llbracket H \rrbracket^{\geq_2}_{[]} = H$$
$$\llbracket H \rrbracket^{\geq_2}_{\sigma;\gamma} = \lceil \text{proj}_1 \, H \rceil^{\geq_2}, \llbracket \text{proj}_2 \, H \rrbracket^{\geq_2}_{\gamma}$$

Finally, we also need lemmas relating the forcing sets and derivability, the *return* (η) and *run* (μ) operations.

$$\eta(-) : \gamma \Vdash_{\overline{s}} \sigma \Rightarrow \gamma \Vdash \sigma \qquad\qquad \mu(-) : \gamma \Vdash \mathbb{N} \Rightarrow \gamma \Vdash_{\overline{s}} \mathbb{N}$$
$$\eta H = \geq_1 \kappa \mapsto \kappa \geq_{\text{refl}} \lfloor H \rfloor_{\geq_1} \qquad\qquad \mu H = H \geq_{\text{refl}} (\geq_1 \alpha \mapsto \alpha)$$

Return shows that we can always lift a member of the strong forcing set to a member of the forcing set. Run shows that, whenever we have a member of a set forcing type $\mathbb{N}$, we can actually obtain a term of System T in normal form from it; note that by definition the sets $\gamma \Vdash_{\overline{s}} \mathbb{N}$ and $\gamma \Vdash_{\overline{r}} \mathbb{N}$ are the same.

The *reify* function $\downarrow (-)$ shows that we can actually run any forcing set, for any type σ, and not just for $\sigma = \mathbb{N}$. It is defined by induction on the type.

$$
\begin{aligned}
&{}^{\gamma}{\downarrow}^{\sigma} (-) : \gamma \Vdash \sigma \Rightarrow \gamma \vDash_{\overline{\Gamma}} \sigma \\
&{}^{\gamma}{\downarrow}^{\mathbb{N}} H = \mu H \\
&{}^{\gamma}{\downarrow}^{\sigma \to \tau} H = \mathsf{lam}({\downarrow}^{\tau} \\
&\qquad\qquad (\geq_1 \kappa \mapsto \\
&\qquad\qquad\qquad H(\geq_1 \cdot \geq_{\mathrm{cons}} \geq_{\mathrm{refl}}) \\
&\qquad\qquad\qquad (\geq_2 \phi \mapsto \\
&\qquad\qquad\qquad\qquad \phi \geq_{\mathrm{refl}} (\lceil^{\sigma;\gamma}{\uparrow}^{\sigma} hyp\rceil^{\geq_2 \cdot \geq_1}) \geq_{\mathrm{refl}} \\
&\qquad\qquad\qquad\qquad (\geq_3 \mapsto \kappa(\geq_3 \cdot \geq_2)))))) \\
&{}^{\gamma}{\downarrow}^{\sigma * \tau} H = \mathsf{pair} \\
&\qquad\qquad {}^{\alpha}{\downarrow}^{\sigma} (\geq_1 \kappa \mapsto \\
&\qquad\qquad\qquad H \geq_1 (\geq_2 \alpha \mapsto \mathrm{proj}_1 \, \alpha \geq_{\mathrm{refl}} (\geq_3 \mapsto \kappa(\geq_3 \cdot \geq_2)))) \\
&\qquad\qquad {}^{\alpha}{\downarrow}^{\tau} (\geq_1 \kappa \mapsto \\
&\qquad\qquad\qquad H \geq_1 (\geq_2 \alpha \mapsto \mathrm{proj}_2 \, \alpha \geq_{\mathrm{refl}} (\geq_3 \mapsto \kappa(\geq_3 \cdot \geq_2))))
\end{aligned}
$$

The case of arrow type forces us to define, at the same time with the reify function, the reflect function $\uparrow (-)$:

$$
\begin{aligned}
&{}^{\gamma}{\uparrow}^{\sigma} (-) : \gamma \vDash_{\mathsf{e}} \sigma \Rightarrow \gamma \Vdash \sigma \\
&{}^{\gamma}{\uparrow}^{\mathbb{N}} p = \eta(\mathsf{e}\, p) \\
&{}^{\gamma}{\uparrow}^{\sigma \to \tau} p = \eta(\geq_2 \alpha \mapsto {\uparrow}^{\tau} \, \mathsf{app}(\llcorner p \lrcorner_{\geq_2}, {}^{\gamma}{\downarrow}^{\sigma} \alpha)) \\
&{}^{\gamma}{\uparrow}^{\sigma * \tau} p = \eta({\uparrow}^{\sigma} \mathsf{fst}\, p, {}^{\gamma}{\uparrow}^{\tau} \mathsf{snd}\, p)
\end{aligned}
$$

Note that this function needs as domain only the neutral terms.

Finally, we are ready to define the *evaluation* function $\llbracket - \rrbracket$ that constructs a member of the forcing set for any input term of System T^+. The definition is by

recursion on the construction of the term p.

$$\gamma[\![(-)]\!]^\sigma_{(-)} : \gamma \vdash \sigma \Rightarrow \forall \gamma' \Vdash \gamma (\gamma' \Vdash \sigma)$$

$$[\![\mathsf{hyp}]\!]_\rho = \mathrm{proj}_1 \rho$$

$$[\![\mathsf{wkn}\, p]\!]_\rho = [\![p]\!]_{\mathrm{proj}_2 \rho}$$

$$[\![\mathsf{lam}\, p]\!]_\rho = \eta(\geq_1 \alpha \mapsto [\![p]\!]_{(\alpha, [\![\rho]\!]^{\geq_1})})$$

$$[\![\mathsf{app}(p,q)]\!]_\rho =_{\geq_1} \kappa \mapsto$$
$$[\![p]\!]_\rho \geq_1$$
$$(\geq_2 \phi \mapsto$$
$$\phi \geq_{\mathrm{refl}} ([\![q]\!]_{[\![\rho]\!]^{\geq_2 \cdot \geq_1}}) \geq_{\mathrm{refl}}$$
$$(\geq_3 \mapsto \kappa(\geq_3 \cdot \geq_2)))$$

$$[\![\mathsf{pair}(p,q)]\!]_\rho = \eta([\![p]\!]_\rho, [\![q]\!]_\rho)$$

$$[\![\mathsf{fst}\, p]\!]_\rho =_{\geq_1} \kappa \mapsto [\![p]\!]_\rho \geq_1 (\geq_2 \alpha \mapsto \mathrm{proj}_1 \alpha \geq_{\mathrm{refl}} (\geq_3 \mapsto \kappa(\geq_3 \cdot \geq_2)))$$

$$[\![\mathsf{snd}\, p]\!]_\rho =_{\geq_1} \kappa \mapsto [\![p]\!]_\rho \geq_1 (\geq_2 \alpha \mapsto \mathrm{proj}_2 \alpha \geq_{\mathrm{refl}} (\geq_3 \mapsto \kappa(\geq_3 \cdot \geq_2)))$$

$$[\![\mathsf{shift}\, p]\!]_\rho =_{\geq_1} \kappa \mapsto \mu [\![p]\!]_{\eta(\geq_2 \nu \mapsto \eta(\geq_3 \alpha \mapsto \eta(\alpha \geq_{\mathrm{refl}}(\geq_4 \mapsto \kappa(\geq_4 \cdot \geq_3 \cdot \geq_2)))))), [\![\rho]\!]^{\geq_1}}$$

$$[\![\mathsf{zero}]\!]_\rho = \eta(\mathsf{zero})$$

$$[\![\mathsf{succ}\, p]\!]_\rho = \eta(\mathsf{succ}\, (\mu [\![p]\!]_\rho))$$

$$[\![\mathsf{rec}(n,a,f)]\!]_\rho =_{\geq_1} \kappa \mapsto [\![n]\!]_\rho \geq_1 (\geq_2 \nu \mapsto \{\nu\}_{[\![\rho]\!]^{\geq_2 \cdot \geq_1}} \geq_{\mathrm{refl}} (\geq_3 \mapsto \kappa(\geq_3 \cdot \geq_2)))$$

where

$$\{\mathsf{zero}\}_{\rho'} = [\![a]\!]_{\rho'}$$

$$\{\mathsf{succ}\, r\}_{\rho'} =_{\geq_1} \kappa \mapsto [\![f]\!]_{[\![\rho']\!]^{\geq_1}} \geq_{\mathrm{refl}} (\geq_2 \gamma \mapsto \gamma \geq_{\mathrm{refl}} (\eta \ulcorner r \urcorner^{\geq_2 \cdot \geq_1}) \geq_{\mathrm{refl}}$$
$$(\geq_3 \delta \mapsto \{r\}_{\rho'}(\geq_3 \cdot \geq_2 \cdot \geq_1)(\geq_4 \alpha \mapsto \delta \geq_4 (\eta\alpha) \geq_{\mathrm{refl}}$$
$$(\geq_5 \mapsto \kappa(\geq_5 \cdot \geq_4 \cdot \geq_3 \cdot \geq_2)))))$$

$$\{\mathsf{e}\, e\}_{\rho'} =\, \uparrow \mathsf{rec}(e, \downarrow [\![a]\!]_{\rho'}, \downarrow [\![f]\!]_{\rho'})$$

Note that the argument ρ is of type $\gamma' \Vdash \gamma$.

For $\gamma' = \gamma$, such a ρ can always be constructed by reflecting the term hyp:

$$\Uparrow \gamma : \gamma \Vdash \gamma$$
$$\Uparrow [] = tt$$
$$\Uparrow \sigma; \gamma = (\uparrow^\sigma \mathsf{hyp}), [\![\Uparrow \gamma]\!]^{\geq_{\mathrm{cons}} \geq_{\mathrm{refl}}}$$

We have therefore shown that, given $p : \gamma \vdash \sigma$ of System T^+, there is a term in normal form $\downarrow [\![p]\!]_\rho : \gamma \models_{\overline{\mathsf{T}}} \sigma$ of System T, for every ρ, and in particular one such term is $\downarrow [\![p]\!]_{\Uparrow \gamma}$. $\square$

The following proposition characterizes the equational theory generated by the normalization procedure. It will be used in the proof of Soundness Theorem of Section 3. This has also been machine checked and is available in Agda notation from [18].

Proposition 1. *The following definitional equalities hold,*

$$\text{(1)} \qquad \downarrow [\![wkn\,p]\!]_{\alpha,\rho} = \downarrow [\![p]\!]_{\rho} \qquad\qquad for\ \alpha \in \tau; \gamma \Vdash \tau$$

$$\text{(2)} \qquad \downarrow [\![hyp]\!]_{\alpha,\rho} = \downarrow \alpha \qquad\qquad for\ \alpha \in \gamma \Vdash \tau$$

$$\text{(3)} \qquad \downarrow [\![fst\,pair(p,q)]\!]_{\rho} = \downarrow [\![p]\!]_{\rho}$$

$$\text{(4)} \qquad \downarrow [\![snd\,pair(p,q)]\!]_{\rho} = \downarrow [\![q]\!]_{\rho}$$

$$\text{(5)} \qquad \downarrow [\![app(lam\,p,q)]\!]_{\rho} = \downarrow [\![p]\!]_{[\![q]\!]_{\rho},\rho}$$

$$\text{(6)} \qquad \downarrow [\![rec(zero,p,q)]\!]_{\rho} = \downarrow [\![p]\!]_{\rho}$$

$$\text{(7)} \qquad \downarrow [\![rec(succ\,r,p,q)]\!]_{\rho} = \downarrow [\![app(app(q,r),rec(r,p,q))]\!]_{\rho}$$

$$\text{(8)} \qquad \downarrow^{\mathbb{N}} [\![shift\,p]\!]_{\rho} = \downarrow^{\mathbb{N}} [\![p]\!]_{\phi,\rho}$$

$$\text{(9)}$$
$$\downarrow^{\mathbb{N}} [\![app(app(hyp,x),y)]\!]_{\phi,\rho} = \downarrow^{\mathbb{N}} [\![y]\!]_{\phi,\rho}$$

where for the last two equations,

$$\phi := \eta(\geq_2 \nu \mapsto \eta(\geq_3 \alpha \mapsto \eta(\mu\alpha))),$$

and $x,y : \mathbb{N} \to \mathbb{N} \to \mathbb{N} \vdash \mathbb{N}$.

Proof. Equations (1)–(7) follow from the ones that hold already of the $[\![-]\!]_{(-)}$ function. This is because, as an argument to the $\downarrow (-)$ function, the evaluation function is always applied to some $\geq_1$ and κ.

$$[\![p]\!]_{\rho} \geq_1 \kappa = [\![q]\!]_{\rho} \geq_1 \kappa$$
$$[\![wkn\,p]\!]_{\alpha,\rho} \geq_1 \kappa = [\![p]\!]_{\rho} \geq_1 \kappa$$
$$[\![hyp]\!]_{\alpha,\rho} \geq_1 \kappa = \alpha \geq_1 \kappa$$
$$[\![fst\,pair(p,q)]\!]_{\rho} \geq_1 \kappa = [\![p]\!]_{\rho} \geq_1 \kappa$$
$$[\![snd\,pair(p,q)]\!]_{\rho} \geq_1 \kappa = [\![q]\!]_{\rho} \geq_1 \kappa$$
$$[\![app(lam\,p,q)]\!]_{\rho} \geq_1 \kappa = [\![p]\!]_{[\![q]\!]_{\rho},\rho} \geq_1 \kappa$$
$$[\![rec(zero,p,q)]\!]_{\rho} \geq_1 \kappa = [\![p]\!]_{\rho} \geq_1 \kappa$$
$$[\![rec(succ\,r,p,q)]\!]_{\rho} \geq_1 \kappa = [\![app(app(q,r),rec(r,p,q))]\!]_{\rho} \geq_1 \kappa$$

These equations come out by unfolding the definition and occasionally using an η-equality step of the form $(\alpha \mapsto \phi\alpha) = \phi$.

Equations (8)–(9) also follow by definition, this time reification being applied for only one concrete type, $\mathbb{N}$. $\qquad\square$

3. A MODIFIED REALIZABILITY INTERPRETATION OF ANALYSIS

By a logical theory sufficient to formalize proofs of Analysis we have in mind the System $HA^{\omega+}$ of Figure 1 together with the full axiom of choice schema $AC = \cup_{\sigma,\tau \in \mathcal{T}} AC^{\sigma\tau}$,

$$(AC^{\sigma\tau}) \qquad\qquad \forall x^{\sigma} \exists^{\tau} y A(x,y) \to \exists^{\sigma\to\tau} f \forall x^{\sigma} A(x,f(x)).$$

This formulation of the axiom is strictly stronger than the Axiom of Dependent Choices which usually treated in the context of realizability interpretations for Analysis. One can also consider the axioms of Figure 2 to be part of $HA^{\omega+}$. It is known that due to their logical form, the proof interpretation that we are going to

$$\frac{}{A, \Gamma \vdash A} \text{ Ax} \qquad\qquad \frac{\Gamma \vdash A}{B, \Gamma \vdash A} \text{ Wkn}$$

$$\frac{A, \Gamma \vdash B}{\Gamma \vdash A \to B} \to_{\text{I}} \qquad\qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B} \to_{\text{E}}$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \wedge_{\text{E}}^{1} \qquad\qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B} \wedge_{\text{E}}^{2}$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \wedge_{\text{I}} \qquad\qquad \frac{\Gamma \vdash A(r^{\tau})}{\Gamma \vdash \exists x^{\tau} A(x)} \exists_{\text{I}}$$

$$\frac{\Gamma \vdash \exists x^{\tau} A(x) \qquad \Gamma \vdash \forall x^{\tau}(A(x) \to B) \qquad x \notin \text{FV}(B)}{\Gamma \vdash B} \exists_{\text{E}}$$

$$\frac{\Gamma \vdash A(x^{\tau}) \qquad x \notin \text{FV}(\Gamma)}{\Gamma \vdash \forall x^{\tau} A(x)} \forall_{\text{I}} \qquad\qquad \frac{\Gamma \vdash \forall x^{\tau} A(x)}{\Gamma \vdash A(r^{\tau})} \forall_{\text{E}}$$

$$\frac{\Gamma \vdash A(\mathsf{zero}) \qquad \Gamma \vdash \forall x^{\mathbb{N}}(A(x) \to A(\mathsf{succ}\, x))}{\Gamma \vdash \forall x^{\mathbb{N}} A(x)} \text{ Ind}$$

$$\frac{\forall x^{\mathbb{N}}(A(x) \to S(x)), \Gamma \vdash S(r)}{\Gamma \vdash A(r)} \text{ Shift} \qquad (A, S \in \Sigma_2)$$

Figure 1: A natural deduction system for the theory $\text{HA}^{\omega+}$

employ has no modifying effect on them. On the other hand, they can indicate in which sense the realizability model supports extensionality.

$\text{HA}^{\omega+}$ is a first-order, predicate logic which is multi-sorted, that is, has variables and quantifiers that range over the types of System T. The system is "minimal" in the sense of Schwichtenberg [28] – one wants to know that the method works even if we do not have a special treatment of the absurdity symbol $\bot$; one can work with any fixed formula N as if it were $\bot$.

$\text{HA}^{\omega+}$ has explicit rules for dealing with the existential quantifier (disjunction has not been included for the sake of simplicity). This, together with the special Shift rule, justifies the plus superscript $^{+}$ in the name. The rule Shift is a general form of the more usual double-negation elimination rule, restricted to Σ_2-formulas, that is precisely suitable for a simple proof of the Soundness Theorem. It has previously been used by Nakata and the author in a semi-classical logic context [16, 19].

The class of Σ_2-*formulas* consists of formulas S of the following form,

$$S ::= N \mid \exists x^{\mathbb{N}} N \mid N \to S \mid N \wedge S \mid S \wedge N,$$

where N stands for so called *computationally irrelevant* formulas [29], defined inductively by

$$N ::= P \mid N \wedge N \mid N \vee N \mid \forall x^{\tau} N \mid A \to N,$$

where P stands for prime formulas (predicates) and A has no restrictions on the form.

$$\frac{\downarrow [\![r]\!]_{\Uparrow \Gamma} = \downarrow [\![s]\!]_{\Uparrow \Gamma}}{\Gamma \vdash r^\tau \doteq s^\tau} \; \text{REFL} \qquad \frac{\Gamma \vdash A(r) \qquad \Gamma \vdash r \doteq s}{\Gamma \vdash A(s)} \; \text{COMP}$$

$$\frac{\Gamma \vdash (\mathsf{succ}\, r) \doteq \mathsf{zero}}{\Gamma \vdash \bot} \; \text{CONT} \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash A} \; \text{EFQ}$$

Figure 2: Additional rules for equality (computationally irrelevant)

As our interpretation we will use a version of Kreisel's so called modified realizability interpretation [33, 20], optimised similarly to the work of Berger, Buchholz, Schwichtenberg, and Seisenberger [2, 29]. However, the key addition is that the realizing terms will also be computed by normalized instances of the shift term.

Definition 1. Given a context Γ and an interpretation of hypotheses $\rho : |\Gamma| \Vdash |\Gamma|$, the *modified realizability interpretation* "p **mr** A" of a formula A by a term p of type $|\Gamma| \vDash |A|$ of System T is defined by the following formula translation,

$$p \; \mathbf{mr} \; N := N \qquad \text{(for any term } p \text{ of type } \mathbb{N})$$
$$p \; \mathbf{mr} \; N \wedge B := N \wedge (p \; \mathbf{mr} \; B)$$
$$p \; \mathbf{mr} \; A \wedge N := (p \; \mathbf{mr} \; A) \wedge N$$
$$p \; \mathbf{mr} \; A \wedge B := (\downarrow [\![\mathsf{fst}\, p]\!]_\rho \; \mathbf{mr} \; A) \wedge (\downarrow [\![\mathsf{snd}\, p]\!]_\rho \; \mathbf{mr} \; B)$$
$$p \; \mathbf{mr} \; N \to B := N \to (p \; \mathbf{mr} \; B)$$
$$p \; \mathbf{mr} \; A \to B := \forall x([\downarrow [\![x]\!]_\rho \; \mathbf{mr} \; A] \to [\downarrow [\![\mathsf{app}(p,x)]\!]_\rho \; \mathbf{mr} \; B])$$
$$p \; \mathbf{mr} \; \forall x^\tau A(x) := \forall x^\tau (\downarrow [\![\mathsf{app}(p,x)]\!]_\rho \; \mathbf{mr} \; A(x))$$
$$p \; \mathbf{mr} \; \exists x^\tau N(x) := N(p)$$
$$p \; \mathbf{mr} \; \exists x^\tau A(x) := \downarrow [\![\mathsf{snd}\, p]\!]_\rho \; \mathbf{mr} \; A(\downarrow [\![\mathsf{fst}\, p]\!]_\rho),$$

in which N denotes a computationally irrelevant formula, and where the type $|A|$ of the realizing term p is computed as follows:

$$|N| := \mathbb{N}$$
$$|N \wedge B| := |B|$$
$$|A \wedge N| := |A|$$
$$|A \wedge B| := |A| * |B|$$
$$|N \to B| := |B|$$
$$|A \to B| := |A| \to |B|$$
$$|\forall x^\tau A| := \tau \to |A|$$
$$|\exists x^\tau N| := \tau$$
$$|\exists x^\tau A| := \tau * |A|$$

The map $|\cdot|$ is extended to contexts Γ by $|C_1, \ldots, C_n| := |C_1|; \cdots ; |C_n|$.

Note that Σ_2-formulae are exactly those ones that are realized by a term of type $\mathbb{N}$.

Our main result is the following one.

Soundness Theorem. *If $HA^{\omega+}+AC$ proves $C_1, C_2, \ldots, C_n \vdash A$, and A is computationally relevant, then there exists a term p of System T^+ such that $HA^{\omega+}$ alone proves that, for every $\rho : |C_1|, |C_2|, \ldots, |C_n| \Vdash |C_1|, |C_2|, \ldots, |C_n|,$*

$$\downarrow \llbracket hyp \rrbracket_\rho \; \boldsymbol{mr} \; C_1, \downarrow \llbracket wkn\,hyp \rrbracket_\rho \; \boldsymbol{mr} \; C_2, \ldots, \downarrow \llbracket wkn^n\,hyp \rrbracket_\rho \; \boldsymbol{mr} \; C_n \vdash \downarrow \llbracket p \rrbracket_\rho \; \boldsymbol{mr} \; A.$$

Proof. The proof is by induction on the derivation of $C_1, C_2, \ldots, C_n \vdash A$ and provides the same realizing terms as Kreisel's modified realizability interpretation of $HA^\omega+AC$. The additional rule of SHIFT (treated in more detail below) is realized via the shift term of System T^+, nevertheless normalized to System T using $\downarrow \llbracket \cdot \rrbracket$.

We will denote by p realizing terms provided by the induction hypothesis for each proof rule. If there are two induction hypotheses, the term corresponding to the second, right premise of the proof rule will be denoted by q.

As a general guide, the elimination rules are enough to prove their own soundness, while the introduction rules and the rules AC, WKN, IND, and SHIFT also need to use the definitional equalities of Proposition 1.

In general, the axiom $AC^{\sigma\tau}$ is realized by the term

$$lam\,pair(lam\,app(fst\,wkn\,hyp, hyp), lam\,app(snd\,wkn\,hyp, hyp)).$$

When the formula $A(x, y)$ is computationally irrelevant, the realizer is the term lam hyp. The proof in both cases is a trivial intuitionistic implication and does not require $AC^{\sigma\tau}$ itself. Equations (1)-(5) of Proposition 1 are nevertheless used.

AX is realized by hyp.

WKN is realized by wkn p and verified using equation (1) and (2).

The general case of $\rightarrow_I$, when A from $A \rightarrow B$ is computationally relevant, is realized by lam p and verified using equations (5), (2), and (1). The induction hypothesis needs to be used with the context $\rho := (\llbracket x \rrbracket_\rho, \rho)$, where x comes from the unfolding of the mr-definition for implication. The special case, when A is computationally irrelevant, is rather realized by the term p only.

When A from $A \rightarrow B$ is computationally relevant, the case $\rightarrow_E$ is realized by $app(p, q)$. When A is irrelevant, the realizing term is just p.

$\wedge_E^1$ is realized using fst p, in general, while in the case where one of the conjuncts of $A \wedge B$ is irrelevant, the realizer is just p.

$\wedge_E^2$ is realized using snd p, in general, while in the case where one of the conjuncts of $A \wedge B$ is irrelevant, the realizer is just p.

In general, $\wedge_I$ is realized using $pair(p, q)$ and verified via equations (3) and (4). When A from $A \wedge B$ is irrelevant, the realizer is q, while when B is irrelevant, the realizer is p.

In general, $\exists_I$ is realized by $pair(p, q)$ and verified via equations (3) and (4). When $A(x)$ from $\exists x A(x)$ is computationally irrelevant, then the realizer is r, the witnessing term.

$\exists_E$ is realized by $app(app(q, fst\,p), snd\,p)$, in general. When $A(x)$ from $\exists x A(x)$ is computationally irrelevant, then the realizer is $app(q, p)$.

$\forall_I$ is realized by lam p. For verification, it is necessary to apply equation (5) and to use the induction hypothesis with context $\rho := (\llbracket x \rrbracket_\rho, \rho)$.

$\forall_E$ is realized by $app(p, r)$.

IND is realized by lam $rec(hyp, p, q)$ and using equations (6) and (7).

Shift is realized by $\mathsf{shift}\,p$ (normalized to System T using $\downarrow[\![-]\!]$). The goal is to prove

$$\downarrow[\![\mathsf{hyp}]\!]_\rho \ \mathbf{mr}\ C_1,\ldots,\downarrow[\![\mathsf{wkn}^n\,\mathsf{hyp}]\!]_\rho \ \mathbf{mr}\ C_n \vdash \downarrow[\![\mathsf{shift}\,p]\!]_\rho \ \mathbf{mr}\ A(r).$$

Using equation (8), we obtain ϕ and the goal becomes

$$\downarrow[\![\mathsf{hyp}]\!]_\rho \ \mathbf{mr}\ C_1,\ldots,\downarrow[\![\mathsf{wkn}^n\,\mathsf{hyp}]\!]_\rho \ \mathbf{mr}\ C_n \vdash \downarrow[\![p]\!]_{\phi,\rho} \ \mathbf{mr}\ A(r).$$

We can now use the induction hypothesis with $\rho := (\phi,\rho)$,

$$\downarrow[\![\mathsf{hyp}]\!]_{\phi,\rho} \ \mathbf{mr}\ \forall x^\mathbb{N}(A(x)\to S(x)),\downarrow[\![\mathsf{wkn}\,\mathsf{hyp}]\!]_{\phi,\rho} \ \mathbf{mr}\ C_1,\ldots,$$
$$\downarrow[\![\mathsf{wkn}^{n+1}\,\mathsf{hyp}]\!]_{\phi,\rho} \ \mathbf{mr}\ C_n \vdash \downarrow[\![p]\!]_{\phi,\rho} \ \mathbf{mr}\ S(r).$$

Thanks to equation (1), the induction hypothesis becomes

$$\downarrow[\![\mathsf{hyp}]\!]_{\phi,\rho} \ \mathbf{mr}\ \forall x^\mathbb{N}(A(x)\to S(x)),\downarrow[\![\mathsf{hyp}]\!]_\rho \ \mathbf{mr}\ C_1,\ldots,$$
$$\downarrow[\![\mathsf{wkn}^n\,\mathsf{hyp}]\!]_\rho \ \mathbf{mr}\ C_n \vdash \downarrow[\![p]\!]_{\phi,\rho} \ \mathbf{mr}\ S(r).$$

Finally, thanks to equation (9), we can finish the proof by applying the Shift rule for:

$$S'(x,y) := \downarrow[\![y]\!]_{\phi,\rho} \ \mathbf{mr}\ S(x)$$
$$A'(x,y) := \downarrow[\![y]\!]_{\phi,\rho} \ \mathbf{mr}\ A(x).$$

$\square$

Remark 1. The Shift case in the proof of the Soundness Theorem only uses the case where shift is reified at type $\mathbb{N}$. This use does not exhaust the possibilities of the realizability model. For example, one can prove the soundness of the Shift rule for $A = A_1 \wedge A_2$ or $A = \exists z^\mathbb{N} A_2(z)$, when $A_i \in \Sigma_2$, by using the equations

$$\downarrow^{\tau*\sigma}[\![p]\!]_\rho = \mathsf{pair}(\downarrow^\tau[\![\mathsf{fst}\,p]\!]_\rho,\ \downarrow^\sigma[\![\mathsf{snd}\,p]\!]_\rho)$$
$$\downarrow^{\mathbb{N}*\sigma}[\![\mathsf{fst}\,\mathsf{shift}\,p]\!]_\rho =\downarrow^\mathbb{N}[\![p]\!]_{\phi_1,\rho}$$
$$\downarrow^{\tau*\mathbb{N}}[\![\mathsf{snd}\,\mathsf{shift}\,p]\!]_\rho =\downarrow^\mathbb{N}[\![p]\!]_{\phi_2,\rho}$$
$$\downarrow^\mathbb{N}[\![\mathsf{app}(\mathsf{app}(\mathsf{hyp},x),y)]\!]_{\phi_1,\rho} =\downarrow^\mathbb{N}[\![\mathsf{fst}\,y]\!]_{\phi_1,\rho}$$
$$\downarrow^\mathbb{N}[\![\mathsf{app}(\mathsf{app}(\mathsf{hyp},x),y)]\!]_{\phi_2,\rho} =\downarrow^\mathbb{N}[\![\mathsf{snd}\,y]\!]_{\phi_2,\rho}$$

where

$$x : \mathbb{N}\to\mathbb{N}*\mathbb{N}\to\mathbb{N};\gamma\vdash\mathbb{N}$$
$$y : \mathbb{N}\to\mathbb{N}*\mathbb{N}\to\mathbb{N};\gamma\vdash\mathbb{N}*\mathbb{N}$$
$$\phi_1 := \eta(\geq_2 \nu \mapsto \eta(\geq_3 \alpha \mapsto \eta(\alpha \geq_\mathrm{refl} (\geq_4 \gamma \mapsto (\mathrm{proj}_1\,\gamma)))))$$
$$\phi_2 := \eta(\geq_2 \nu \mapsto \eta(\geq_3 \alpha \mapsto \eta(\alpha \geq_\mathrm{refl} (\geq_4 \gamma \mapsto (\mathrm{proj}_2\,\gamma))))).$$

The induction hypothesis needs to be used twice, once for ϕ_1 and once for ϕ_2.

Similar equations hold for function types:

$$^\gamma\downarrow^\mathbb{N}[\![\mathsf{app}(\mathsf{shift}\,p,z)]\!]_\rho =^{\mathbb{N}\to(\mathbb{N}\to\mathbb{N})\to\mathbb{N};\gamma}\downarrow^\mathbb{N}[\![p]\!]_{\phi_3,\rho}$$
$$\phi_3 := \eta(\geq_2 \nu \mapsto \eta(\geq_3 \alpha \mapsto \eta(\alpha \geq_\mathrm{refl} (\geq_4 \gamma \mapsto \mu(\gamma \geq_\mathrm{refl} [\![z]\!]_{\lceil\rho\rceil\geq_4 \cdot \geq_3 \cdot \geq_2})))))$$

Nevertheless, it does not appear to be possible to prove the soundness of the Shift rule even for the case $|A| = \mathbb{N} \to \mathbb{N}$ in general, since it is well known that there are already classically true Σ_3^0-formulas which do not have a recursive realizer.

Still, it may be the case that the realizability model can be used to give a sound computational interpretation of *particular* Σ_3^0 (or more complex) formulas.

Corollary 1. *The Σ_2-fragment of classical Analysis satisfies the Existence Property,*

> *Given a derivation of $\Gamma \vdash \exists x^\tau A(x)$, there exists a term p of type τ of System T such that $\Gamma \vdash A(p)$.*

and, consequently, the Weak Church's Rule,

> *Given a (closed) derivation of $\emptyset \vdash \forall x^\mathbb{N} \exists y^\mathbb{N} A(x,y)$, there exists a total recursive function $\mathbf{f} : \mathbb{N} \to \mathbb{N}$ such that, for all $\mathbf{n} \in \mathbb{N}$, we have that $\emptyset \vdash A(\overline{\mathbf{n}}, \overline{\mathbf{fn}})$, where $\overline{\mathbf{m}}$ denotes the term $\underbrace{\mathsf{succ} \cdots \mathsf{succ}}_{\mathbf{m}\ times} \mathsf{zero}.$*

Proof. The proof method is not new (see Corollary 5.24 of [20] and paragraph 1.11.7 of [33]). If the formula A is of the class Γ_1 [33, 20],

$$\Gamma_1 \ni G ::= N \mid G \wedge G \mid \forall x G \mid \exists x G \mid S \to G,$$

(where S is a Σ_2-formula and N is computationally irrelevant) then already intuitionistic logic shows that $\vdash (p\ \mathbf{mr}\ A) \to A$. If A is outside this class, one first needs to define the "with truth" variant of modified realizability in which one replaces the clause for implication of mr-interpretation of Definition 1 by

$$p\ \mathbf{mrt}\ A \to B := \forall x([\downarrow [\![x]\!]_\rho\ \mathbf{mrt}\ A] \to [\downarrow [\![\mathsf{app}(p,x)]\!]_\rho\ \mathbf{mrt}\ B]) \wedge (A \to B).$$

The Soundness Theorem is provable for mrt with the same realizing terms, but now we also have, for any formula A, $(p\ \mathbf{mrt}\ A) \to A$. This directly implies the Existence Property.

For the special case when $\Gamma = \emptyset$ and $\tau = \mathbb{N}$, we get the Numerical Existence Property:

> Given a (closed) derivation of $\emptyset \vdash \exists y^\mathbb{N} A(y)$, there exists $\mathbf{n} : \mathbb{N}$ such that $\emptyset \vdash A(\overline{\mathbf{n}})$.

This follows from the fact that a closed derivation has a realizer that is a closed term (does not have non-bound hyp subterms). Since the realizer is necessarily in normal form, and since it is not neutral (all neutral terms have at least one non-bound occurrence of hyp), then the realizer must be of the required form $\underbrace{\mathsf{succ} \cdots \mathsf{succ}}_{\mathbf{n}\ times} \mathsf{zero}$.

To show the Weak Church's Rule, we use elementary Recursion Theory. Like all theories over countable languages, $\mathrm{HA}^{\omega+}+\mathrm{AC}$ is recursively axiomatizable, that is, there exists a recursive predicate $\mathrm{Proof}(k,l)$ formalizing the fact that $k \in \mathbb{N}$ is a code for a derivation of the formula coded by $l \in \mathbb{N}$.

Let $g(n) = \min_m \mathrm{Proof}(j_1 m, \ulcorner A(\overline{n}, \overline{j_2 m}) \urcorner)$, where j_1 and j_2 are the projections of some surjective pairing function. As defined, g is a partial recursive function.

Now, given $\emptyset \vdash \forall x^\mathbb{N} \exists y^\mathbb{N} A(x,y)$ and $n \in \mathbb{N}$, we obtain $\emptyset \vdash \exists y^\mathbb{N} A(\overline{n}, y)$, and by the Numerical Existence Property we obtain $m \in \mathbb{N}$ such that $\vdash A(\overline{n}, \overline{m})$. We proved that, for every n, there exists m such that $\emptyset \vdash A(\overline{n}, \overline{m})$ which shows that the function g is *total* recursive. We may now take $f(n) := j_2(g(n))$ and by definition we have that, for any n, $\emptyset \vdash A(\overline{n}, \overline{f(n)})$. $\qquad\square$

Note that the class Σ_2 includes the following schemata,

$$(\mathrm{MP}) \qquad\qquad\qquad \neg_N \neg_N \exists x^\tau M(x) \to \exists x^\tau M(x),$$

$$\text{(DNS)} \qquad \forall x^{\mathbb{N}} \neg_N \neg_N A(x) \to \neg_N \neg_N \forall x^{\mathbb{N}} A(x),$$

where M, N denote computationally irrelevant formulas and $\neg_N A$ denotes negation in minimal logic, that is $A \to N$ for a fixed N.

The Existence Property implies that principles like MP and DNS can justly be considered as constructive even in presence of induction and the full Axiom of Choice, partly extending previous works [33, 8, 30, 12, 16]. Similar conclusions follow from the work of Rand Moschovakis that uses a version of Kleene's general-recursive realizability [26].

Weak Church's Rule seems to justify why, even constructively, CT_0 deserves the name "the false Church's Thesis" [25].

Acknowledgments

References

[1] Ulrich Berger. A computational interpretation of open induction. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (LICS'04)*, pages 326–334. IEEE Computer Society, 2004.

[2] Ulrich Berger, Wilfried Buchholz, and Helmut Schwichtenberg. Refined program extraction from classical proofs. *Annals of Pure and Applied Logic*, 114:3–25, 2002.

[3] Ulrich Berger and Paulo Oliva. Modified bar recursion and classical dependent choice. *Springer-Verlag Lecture Notes in Logic*, 20:89–107, 2005.

[4] Ulrich Berger and Paulo Oliva. Modified bar recursion. *Mathematical Structures in Computer Science*, 16:163–183, 2006.

[5] Ulrich Berger and Helmut Schwichtenberg. An inverse of the evaluation functional for typed lambda-calculus. In *LICS*, pages 203–211. IEEE Computer Society, 1991.

[6] Olivier Danvy. Type-directed partial evaluation. In *POPL*, pages 242–257, 1996.

[7] Olivier Danvy and Andrzej Filinski. Abstracting control. In *LISP and Functional Programming*, pages 151–160, 1990.

[8] Dov M. Gabbay. Applications of trees to intermediate logics. *The Journal of Symbolic Logic*, 37:135–138, 1972.

[9] Kurt Gödel. *Collected works. Publications 1938–1974*, volume II, chapter Postscript to Spector 1962, page 253. The Clarendon Press Oxford University Press, New York, 1962.

[10] Kurt Gödel. *Collected works. Publications 1938–1974*, volume II, chapter On a hitherto unutilized extension of the finitary standpoint, pages 241–251. The Clarendon Press Oxford University Press, New York, 1990. (English translation of the original 1958 article).

[11] Kurt Gödel. *Collected works. Unpublished essays and lectures*, volume III, chapter In what sense is intuitionistic logic constructive, pages 189–200. The Clarendon Press Oxford University Press, 1995. (early lecture on the Dialectica interpretation from 1941).

[12] Hugo Herbelin. An intuitionistic logic that proves Markov's principle. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pages 50–56. IEEE Computer Society, 2010.

[13] Hugo Herbelin. A constructive proof of dependent choice, compatible with classical logic. In *Proceedings of the 27th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2012, 25-28 June 2012, Dubrovnik, Croatia*, pages 365 – 374. IEEE Computer Society, 2012.

[14] William Alvin Howard. Functional interpretation of bar induction by bar recursion. *Composition Mathematica*, 20:107–124, 1968.

[15] Danko Ilik. *Constructive Completeness Proofs and Delimited Control*. PhD thesis, École Polytechnique, October 2010.

[16] Danko Ilik. Delimited control operators prove double-negation shift. *Annals of Pure and Applied Logic*, 163(11):1549 – 1559, 2012.

[17] Danko Ilik. Continuation-passing style models complete for intuitionistic logic. *Annals of Pure and Applied Logic*, 164(6):651 – 662, 2013.

[18] Danko Ilik. Formal proof of normalization of System T+ in Agda. Available at `http://www.lix.polytechnique.fr/~danko/shift-analysis.zip`, 2014.

[19] Danko Ilik and Keiko Nakata. A direct version of Veldman's proof of open induction on Cantor space via delimited control operators. *Leibniz International Proceedings in Informatics (LIPIcs)*, 26:188–201, 2014.

[20] Ulrich Kohlenbach. *Applied Proof Theory: Proof Interpretations and Their Use in Mathematics*. Springer Monographs in Mathematics. Springer-Verlag, Berlin, Heidelberg, 2008.

[21] Georg Kreisel. Interpretation of analysis by means of constructive functionals of finite types. In Arend Heyting, editor, *Constructivity in Mathematics, Proceedings of the colloqium held at Amsterdam, 1957*, Studies in Logic and The Foundations of Mathematics, pages 101–127. North-Holland Publishing Company Amsterdam, 1959.

[22] Georg Kreisel. Review of the paper "The model G of the theory BR" by Ersov. *Zentralblatt für Mathematik und ihre Grenzgebiete*, 312, 1976.

[23] Jean-Louis Krivine. Dependent choice, 'quote' and the clock. *Theoretical Computer Science*, 308(1–3):259 – 276, 2003.

[24] Sigekatu Kuroda. Intuitionistische untersuchungen der formalistischer logik. *Nagoya Mathematical Journal*, 3:35–47, 1951.

[25] Henri Lombardi and Claude Quitté. *Algèbre commutative – Méthodes constructives*. Calvage & Mounet, Paris, 2011.

[26] Joan Rand Moschovakis. Analyzing realizability by Troelstra's methods. *Annals of Pure and Applied Logic*, 114:203–225, 2002.

[27] Helmut Schwichtenberg. On bar recursion of types 0 and 1. *The Journal of Symbolic Logic*, 44(3), 1979.

[28] Helmut Schwichtenberg and Stanley S. Wainer. *Proofs and Computations*. Perspectives in Logic. Cambridge University Press, 2012.

[29] Monika Seisenberger. Program from proofs using classical dependent choice. *Annals of Pure and Applied Logic*, 153:97–110, 2008.

[30] Jonathan P. Seldin. On the proof theory of the intermediate logic MH. *The Journal of Symbolic Logic*, 51(3):626–647, 1986.

[31] Clifford Spector. Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles formulated in current intuitionistic mathematics. In *Proc. Sympos. Pure Math., Vol. V*, pages 1–27. American Mathematical Society, Providence, R.I., 1962.

[32] Marc Bezem Stefano Berardi and Thierry Coquand. On the computational content of the axiom of choice. *The Journal of Symbolic Logic*, 63(2):600–622, 1998.

[33] Anne S. Troelstra, editor. *Metamathematical Investigations of Intuitionistic Arithmetic and Analysis*. Number 344 in Lecture Notes in Mathematics. Springer-Verlag, Berlin, Heidelberg, New York, 1973.

[34] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013.