CHAPTER ??

Realizability

A. S. Troelstra[1]
Faculteit Wiskunde en Informatica
Universiteit van Amsterdam
Plantage Muidergracht 24
1018TV AMSTERDAM (NL)

Contents

[1] S. Buss, U. Kohlenbach, H. Luckhardt, J.R. Moschovakis and J. van Oosten have commented on earlier drafts of this paper. Van Oosten also provided a sketch for section 8 which has been used in composing the final version.

1 Numerical realizability

1.1. *Introduction*

There is not just one single notion of realizability, but a whole family of notions, which of course resemble each other in certain respects. This section is devoted to a fairly detailed discussion of the earliest and most basic notion of realizability, S.C. Kleene'sKleene, S.C. realizability by numbers. In later sections we discuss more briefly variations of the basic notion. We do not aim at an exhaustive description of all possible proof-theoretic applications of realizability, but rather aim at presenting illustrative examples. Most of the sections are followed by "Notes", containing suggestions for further reading, some historical comments, etc. The historical comments concern mainly the period *after* 1972, since the history up till 1972 is fairly completely documented in (**?**).

Realizability by numbers was introduced in (**?**) as a semantics for intuitionistic arithmetic, by defining for arithmetical sentences A a notion "the number $\mathbf{n}$ realizes A", intended to capture some essential aspects of the intuitionistic meaning of A. Here $\mathbf{n}$ is not a term of the arithmetical formalism, but an element of the natural numbers $\mathbb{N}$. The definition is by induction on the complexity of A:

- $\mathbf{n}$ realizes $t = s$ iff $t = s$ holds;
- $\mathbf{n}$ realizes $A \wedge B$ iff $\mathbf{p}_0\mathbf{n}$ realizes A and $\mathbf{p}_1\mathbf{n}$ realizes B;
- $\mathbf{n}$ realizes $A \vee B$ iff $\mathbf{p}_0\mathbf{n} = 0$ and $\mathbf{p}_1\mathbf{n}$ realizes A or $\mathbf{p}_0\mathbf{n} = 1$ and $\mathbf{p}_1\mathbf{n}$ realizes B;
- $\mathbf{n}$ realizes $A \to B$ iff for all $\mathbf{m}$ realizing A, $\mathbf{n}\bullet\mathbf{m}$ is defined and realizes B;
- $\mathbf{n}$ realizes $\neg A$ if for no $\mathbf{m}$, $\mathbf{m}$ realizes A;
- $\mathbf{n}$ realizes $\exists y\, A$ iff $\mathbf{p}_1\mathbf{n}$ realizes $A[y/\overline{\mathbf{p}_0\mathbf{n}}]$.
- $\mathbf{n}$ realizes $\forall y\, A$ iff $\mathbf{n}\bullet\mathbf{m}$ is defined and realizes $A[y/\overline{\mathbf{m}}]$, for all $\mathbf{m}$.

Here $\mathbf{p}_1$ and $\mathbf{p}_0$ are the inverses of some standard primitive recursive pairing function $\mathbf{p}$ coding $\mathbb{N}^2$ onto $\mathbb{N}$, and $\overline{\mathbf{m}}$ is the standard term $S^{\mathbf{m}}0$ (numeral) in the language of intuitionistic arithmetic corresponding to $\mathbf{m}$; $\bullet$ is partial recursive function application, i.e. $\mathbf{n}\bullet\mathbf{m}$ is the result of applying the function with code $\mathbf{n}$ to $\mathbf{m}$. (Later on we also use $\bar{m}, \bar{n}, \ldots$ for numerals.)

The definition may be extended to formulas with free variables by stipulating that $\mathbf{n}$ realizes A if $\mathbf{n}$ realizes the universal closure of A.

Reading "there is a number realizing A" as "A is constructively true", we see that a realizing number provides witnesses for the constructive truth of existential quantifiers and disjunctions, and in implications carries this type of information from premise to conclusion by means of partial recursive operators. In short, realizing numbers "hereditarily" encode information about the realization of existential quantifiers and disjunctions.

Realizability, as an interpretation of "constructively true" is reminiscent of the well-known BrouwerBrouwer, L.E.J.-HeytingHeyting, A.-KolmogorovKolmogorov, A. explanation (BHK for short) of the intuitionistic meaning of the logical connectives. BHK explains "p proves A" for compound A in terms of the provability of the components of A. For prime formulas the notion of proof is supposed to be given. Examples of the clauses of BHK are:

- p proves $A \to B$ iff p is a construction transforming any proof c of A into a proof $p(c)$ of B;
- p proves $A \wedge B$ iff $p = (p_0, p_1)$ and p_0 proves A, p_1 proves B;
- p proves $A \vee B$ iff $p = (p_0, p_1)$ with $p_0 \in \{0, 1\}$, and p_1 proves A if $p_0 = 0$, p_1 proves B if $p_0 \neq 0$.

Realizability corresponds to BHK if (a) we concentrate on (numerical) information concerning the realizations of existential quantifiers and the choices for disjunctions, and (b) the constructions considered for $\forall, \to$ are encoded by (partial) recursive operations.

Realizability gives a classically meaningful definition of intuitionistic truth; the set of realizable statements is closed under deduction and must be consistent, since 1=0 cannot be realizable. It is to be noted that decidedly non-classical principles are realizable, for example

$$\neg\forall x[\exists y Txxy \vee \forall y \neg Txxy]$$

is easily seen to be realizable. (T is Kleene'sKleene, S.C. T-predicate, which is assumed to be available in our language; $Txyz$ is primitive recursive in x, y, z and expresses that the algorithm with code x applied to argument y yields a computation with code z; U is a primitive recursive function extracting from a computation code z the result Uz.) For $\neg A$ is realizable iff no number realizes A, and realizability of $\forall x[\exists y Txxy \vee \forall y \neg Txxy]$ requires a total recursive function deciding $\exists y Txxy$, which does not exist (more about this below). In this way realizability shows how in constructive mathematics principles may be incorporated which cause it to diverge from the corresponding classical theory, instead of just being included in the classical theory.

Some notational habits adopted in this paper are: dropping of distinguishing sub- and superscripts where the context permits; saving on parentheses, e.g. for a binary predicate R applied to x, y we often write Rxy instead of $R(x, y)$ (this habit has just been demonstrated above). The symbol $\equiv$ is used for literal identity of expressions modulo renaming of bound variables. $\Rightarrow$ is used as metamathematical consequence relation, and in particular $\mathcal{A}, \mathcal{B} \Rightarrow \mathcal{C}$ expresses a rule which derives $\mathcal{C}$ from premises $\mathcal{A}, \mathcal{B}$. $\mathrm{FV}(\mathcal{A})$ is the set of free variables of expression $\mathcal{A}$.

1.2. *Formalizing realizability in* **HA**

In order to exploit realizability proof-theoretically, we have to formalize it. Let us first discuss its formalization in ordinary intuitionistic first-order arithmetic **HA** ("Heyting's Arithmetic"), based on intuitionistic predicate logic with equality, and containing symbols for all primitive recursive functions, with their recursion equations as axioms.

$x, y, z, \ldots$ are numerical variables, S is successor. We use the notation $\bar{n}$ for the term $S^n 0$; such terms are called *numerals*. $\mathbf{p}_0, \mathbf{p}_1$ bind stronger than infix binary operations, i.e. $\mathbf{p}_0 t + s$ is $(\mathbf{p}_0 t) + s$. For primitive recursive predicates R, $Rt_1 \ldots t_n$ may be treated as a prime formula since the formalism contains a symbol for the characteristic function χ_R.

Now we are ready for a formalized definition of "x realizes A" in **HA**.

DEFINITION. By recursion on the complexity of A we define $x \,\underline{\mathtt{rn}}\, A$, $x \notin \mathrm{FV}(A)$, "x numerically realizes A" :

$$\begin{array}{ll}
x \,\underline{\mathtt{rn}}\, (t = s) & := (t = s) \\
x \,\underline{\mathtt{rn}}\, (A \wedge B) & := (\mathbf{p}_0 x \,\underline{\mathtt{rn}}\, A) \wedge (\mathbf{p}_1 x \,\underline{\mathtt{rn}}\, B), \\
x \,\underline{\mathtt{rn}}\, (A \to B) & := \forall y (y \,\underline{\mathtt{rn}}\, A \to \exists z (Txyz \wedge Uz \,\underline{\mathtt{rn}}\, B)), \\
x \,\underline{\mathtt{rn}}\, \forall y\, A & := \forall y \exists z (Txyz \wedge Uz \,\underline{\mathtt{rn}}\, A), \\
x \,\underline{\mathtt{rn}}\, \exists y\, A & := \mathbf{p}_1 x \,\underline{\mathtt{rn}}\, A[y/\mathbf{p}_0 x].
\end{array}$$

Note that $\mathrm{FV}(x \,\underline{\mathtt{rn}}\, A) \subset \{x\} \cup \mathrm{FV}(A)$. □

REMARKS. (i) We have omitted clauses for negation and disjunction, since in arithmetic we can take $\neg A := A \to 1 = 0$, $A \vee B := \exists x((x = 0 \to A) \wedge (x \neq 0 \to B))$. If we spell out $x \,\underline{\mathtt{rn}}\, (A \vee B)$ on the basis of this definition, we find

$$x \,\underline{\mathtt{rn}}\, (A \vee B) := (\mathbf{p}_0 x = 0 \to (\mathbf{p}_0 \mathbf{p}_1 x) 0 \,\underline{\mathtt{rn}}\, A) \wedge (\mathbf{p}_0 x \neq 0 \to (\mathbf{p}_1 \mathbf{p}_1 x) 0 \,\underline{\mathtt{rn}}\, B),$$

(ii) The definition of realizability permits slight variations, e.g. for the first clause we might have taken

$$x \,\underline{\mathtt{rn}}'(t = s) := (x = t \wedge t = s).$$

However, it is routine to see that this variant $\underline{\mathtt{rn}}'$-realizability is *equivalent* to $\underline{\mathtt{rn}}$-realizability in the following sense: for each formula A there are two partial recursive functions ϕ_A and ψ_A such that

$$\begin{array}{l}
\vdash x \,\underline{\mathtt{rn}}\, A \to \phi_A(x) \,\underline{\mathtt{rn}}'A \\
\vdash x \,\underline{\mathtt{rn}}'A \to \psi_A(x) \,\underline{\mathtt{rn}}\, A.
\end{array}$$

(If in the future we shall call two versions of a realizability notion equivalent, it will always be in this or a similar sense.) Similarly, if we treat $\vee$ as a primitive, the clause for $x\,\underline{\mathtt{rn}}\,(A \vee B)$ given above may be simplified to

$$x\,\underline{\mathtt{rn}}\,(A \vee B) := (\mathbf{p}_0 x = 0 \wedge \mathbf{p}_1 x\,\underline{\mathtt{rn}}\,A) \vee (\mathbf{p}_0 x \neq 0 \wedge \mathbf{p}_1 x\,\underline{\mathtt{rn}}\,B),$$

which yields an equivalent notion of realizability.

(iii) In terms of partial recursive function application $\bullet$ and the definedness predicate $\downarrow$ ($t\downarrow$ means "t is defined"), we can write more succinctly:

$$\begin{aligned} x\,\underline{\mathtt{rn}}\,(A \to B) &:= \forall y(y\,\underline{\mathtt{rn}}\,A \to x{\bullet}y\downarrow \wedge\, x{\bullet}y\,\underline{\mathtt{rn}}\,B), \\ x\,\underline{\mathtt{rn}}\,\forall y\, A &:= \forall y(x{\bullet}y\downarrow \wedge\, x{\bullet}y\,\underline{\mathtt{rn}}\,B). \end{aligned}$$

where $t\downarrow$ expresses that t is defined (cf. next subsection). Of course, the partial operation $\bullet$ and the definedness predicate $\downarrow$ are not part of the language, but expressions containing them may be treated as abbreviations, using the following equivalences:

$$\begin{aligned} &t_1 = t_2 \leftrightarrow \exists x(t_1 = x \wedge t_2 = x), \\ &t_1{\bullet}t_2 = x \leftrightarrow \exists yzu(t_1 = y \wedge t_2 = z \wedge Tyzu \wedge Uu = x), \\ &t\downarrow \leftrightarrow \exists z(t = z). \end{aligned}$$

(t_1, t_2 terms containing $\bullet$, x, y, z, u not free in t_1, t_2). However, note that the logical complexity of $A(t)$, where t is an expression containing $\bullet$, depends on the complexity of t! (On the other hand, $t\downarrow$ is always expressible in Σ^0_1-form.) For metamathematical investigations it is therefore more convenient to formalize realizability in a conservative extension $\mathbf{HA}^*$ of $\mathbf{HA}$ in which we can treat "$\bullet$" as a primitive. Treating $t_1 = t_2$ for partially defined t_1, t_2 as an abbreviation in a rigorous way is possible, but involves a good deal of lengthy inductions, as demonstrated in (**?**). Since ordinary logic deals with total functions only, we first need to extend our logic to the (intuitionistic) logic of partial terms LPT, or intuitionistic E^+-logic, in the terminology of Troelstra and van Dalen(**?**, 2.2.3). LPT first appeared in (**?**).

1.3. *Intuitionistic predicate logic with partial terms LPT*

Variables are supposed to range over the objects of the domain considered, so always denote; arbitrary terms need not denote, so we need a predicate **E**, expressing definedness; **E**t reads "t denotes" or "t is defined". Instead of **E**t we shall write $t\downarrow$, in the notation commonly used in recursion theory.

If we also have equality in our logic, and read $t = s$ as "t and s are both defined and equal", we can express $t\downarrow$ as $t = t$.

The following axiomatization is a convenient (but not canonical) choice for arguments proceeding by induction on the length of formal deductions:

L1	$A \to A$,
L2	$A,\ A \to B \Rightarrow B$,
L3	$A \to B,\ B \to C \Rightarrow A \to C$,
L4	$A \land B \to A,\ A \land B \to B$,
L5	$A \to B,\ A \to C \Rightarrow A \to B \land C$,
L6	$A \to A \lor B,\ B \to A \lor B$,
L7	$A \to C,\ B \to C \Rightarrow A \lor B \to C$,
L8	$A \land B \to C \Rightarrow A \to (B \to C)$,
L9	$A \to (B \to C) \Rightarrow A \land B \to C$,
L10	$\bot \to A$,
L11	$B \to A \Rightarrow B \to \forall x\, A\ \ (x \notin \mathrm{FV}(B))$,
L12	$\forall x\, A \land t{\downarrow} \to A[x/t]$,
L13	$A[x/t] \land t{\downarrow} \to \exists x\, A$,
L14	$A \to B \Rightarrow \exists x\, A \to B\ \ (x \notin \mathrm{FV}(B))$

where $t{\downarrow} := t = t$. For equality we have (F function symbol, R relation symbol of the language):

EQ
$$\begin{cases} \forall xy(x = y \to y = x), \quad \forall xyz(x = y \land y = z \to x = z), \\ \forall \vec{x}\vec{y}(\vec{x} = \vec{y} \land F\vec{x}{\downarrow} \to F\vec{x} = F\vec{y}), \quad \forall \vec{x}\vec{y}(R\vec{x} \land \vec{x} = \vec{y} \to R\vec{y}) \end{cases}$$

Basic predicates and functions of the language are assumed to be strict:

STR $\quad F(t_1, \ldots, t_n){\downarrow} \to t_i{\downarrow}, \quad R(t_1, \ldots, t_n) \to t_i{\downarrow}$

Note that this logic reduces to ordinary first-order intuitionistic logic if all functions are total, i.e. $\forall \vec{x}(f\vec{x}{\downarrow})$, since then $t{\downarrow}$ for all terms t.

For the notion *"equally defined and equal if defined"* introduced by

$$t \simeq s := (t{\downarrow} \lor s{\downarrow}) \to t = s,$$

we can prove the replacement schema for arbitrary formulas A

$$t \simeq s \land A[x/t] \to A[x/s].$$

1.4. *Conservativeness of defined functions*

Relative to the logic of partial terms, the following conservative extension result is easily proved. Let Γ be a theory based on LPT, such that

$$\Gamma \vdash A(\vec{x}, y) \land A(\vec{x}, z) \to y = z.$$

Then we may introduce a symbol ϕ_A for a partial function with axiom

$\mathrm{Ax}(\phi_A) \qquad A(\vec{x}, y) \leftrightarrow y = \phi_A(\vec{x})$.

The conservativeness of this addition can be proved in a straightforward syntactic way; the easiest method, however, uses completeness for Kripke models, see (**?**, 2.7).

Let Γ^* consist of Γ and all substitution instances of the axiom schemata w.r.t. the extended language, and let $\phi(\Gamma^*)$ be the result of systematically eliminating the function symbol ϕ_A from the elements of Γ, and assume $\phi(\Gamma^*)$ to be provable from Γ, then the conservative extension result still holds in the form: "$\Gamma^* + \mathrm{Ax}(\phi_A)$ is conservative over Γ".

This extended result applies to $\mathbf{HA}^*$ defined below, since eliminating the symbol for partial recursive function application from instances of induction yields instances of induction in the language of $\mathbf{HA}$.

1.5. *Formalizing elementary recursion theory in* $\mathbf{HA}^*$

$\mathbf{HA}^*$ is the conservative extension of $\mathbf{HA}$, formulated in the intuitionistic logic of partial terms, with a primitive binary partial operation $\bullet$ of partial recursive function application. $t_1 \bullet t_2 \bullet t_3 \ldots$ abbreviates $(\ldots((t_1 \bullet t_2) \bullet t_3)\ldots)$ (association to the left).

Note that strictness entails in particular $t \bullet t' \downarrow \to t\downarrow \wedge t'\downarrow$ for the application operation. Of course we have to require totality for the primitive recursive functions; it suffices to demand $0\downarrow$, $Sx\downarrow$. In all other cases the primitive recursive functions satisfy equations with $=$, characterizing them inductively in terms of functions introduced before (e.g. $x + 0 = x$, $x + Sy = S(x + y)$). By induction one can then prove $Fx_1 \ldots x_n\downarrow$ for each primitive recursive function symbol F.

A smooth formalization of elementary recursion theory in $\mathbf{HA}^*$ can be given by using Kleene'sKleene, S.C. index method in combination with the theory of elementary inductive definitions in arithmetic (**?**, 3.6, 3.7). In particular we obtain the smn-theorem, the recursion theorem (Kleene'sKleene, S.C. fixed-point theorem), the KleeneKleene, S.C. normal form theorem, etc. Moreover, by the normal form theorem, every partial recursive function is definable by a term of the language of $\mathbf{HA}^*$.

NOTATION. If t is a term in the language of $\mathbf{HA}^*$, then $\Lambda x.t$ is a canonically chosen code number for t as a partial recursive function of x, uniformly in the other free variables; by the smn-theorem we may therefore assume $\Lambda x.t$ to be primitive recursive in $\mathrm{FV}(t) \setminus \{x\}$. $\Lambda x_1 \ldots x_n.t$ abbreviates $\Lambda x_1(\Lambda x_2 \ldots (\Lambda x_n.t)\ldots)$. $\Box$

We note the following

LEMMA. *In* $\mathbf{HA}^*$ *the* Σ^0_1*-formulas of* $\mathbf{HA}$ *are equivalent to prime formulas of the form* $t = t$ *for suitable* t*, and each formula* $t = s$ *is equivalent to a* Σ^0_1*-formula of* $\mathbf{HA}$.

Proof. Systematically using the equivalences mentioned above transforms any formula $t = s$ of $\mathbf{HA}^*$ into a Σ^0_1-formula of $\mathbf{HA}$. Conversely, let a Σ^0_1-formula be given; by the normal form results of recursion theory, we can write this in the form $\exists z T(\bar{n}, \langle \vec{x} \rangle, z)$ for a numeral $\bar{n}$; this is equivalent to $\bar{n} \bullet \langle \vec{x} \rangle = \bar{n} \bullet \langle \vec{x} \rangle$. □

We are now ready to formalize $x \,\underline{\mathtt{rn}}\, A$ directly in $\mathbf{HA}^*$.

1.6. *Formalizing* $\underline{\mathtt{rn}}$*-realizability in* $\mathbf{HA}^*$

DEFINITION. $x \,\underline{\mathtt{rn}}\, A$ is defined by induction on the complexity of A, $x \notin \mathrm{FV}(A)$.

$$
\begin{array}{ll}
x \,\underline{\mathtt{rn}}\, P & := P \wedge x{\downarrow} \text{ for } P \text{ prime},\\
x \,\underline{\mathtt{rn}}\, (A \wedge B) & := \mathbf{p}_0 x \,\underline{\mathtt{rn}}\, A \wedge \mathbf{p}_1 x \,\underline{\mathtt{rn}}\, B,\\
x \,\underline{\mathtt{rn}}\, (A \to B) & := \forall y (y \,\underline{\mathtt{rn}}\, A \to x \bullet y \,\underline{\mathtt{rn}}\, B) \wedge x{\downarrow},\\
x \,\underline{\mathtt{rn}}\, \forall y\, A & := \forall y (x \bullet y \,\underline{\mathtt{rn}}\, A),\\
x \,\underline{\mathtt{rn}}\, \exists y\, A & := \mathbf{p}_1 x \,\underline{\mathtt{rn}}\, A[y/\mathbf{p}_0 x].
\end{array}
$$

We also define a combination of realizability with truth, $x \,\underline{\mathtt{rnt}}\, A$; the clauses are the same as for $\underline{\mathtt{rn}}$, the clause for implication excepted, which now reads:

$$x \,\underline{\mathtt{rnt}}\, (A \to B) := \forall y (y \,\underline{\mathtt{rnt}}\, A \to x \bullet y \,\underline{\mathtt{rnt}}\, B) \wedge x{\downarrow} \wedge (A \to B). \qquad \square$$

REMARKS. (i) $t \,\underline{\mathtt{rn}}\, A$ is $\exists$-free (i.e. does not contain $\exists$) for all A. Note that, by our definition of $\vee$ in terms of the other operators, $\exists$-free implies $\vee$-free.

(ii) The clauses "$\wedge\, x{\downarrow}$" have been added for the cases of prime formulas and implications, in order to guarantee the truth of part (i) of the following lemma.

(iii) For negations we have $x \,\underline{\mathtt{rn}}\, \neg A \leftrightarrow \forall y (\neg y \,\underline{\mathtt{rn}}\, A) \wedge x{\downarrow}$, and $x \,\underline{\mathtt{rn}}\, \neg\neg A \leftrightarrow \forall y (\neg y \,\underline{\mathtt{rn}}\, \neg A) \wedge x{\downarrow} \leftrightarrow \forall y \neg \forall z \neg (z \,\underline{\mathtt{rn}}\, A) \wedge x{\downarrow} \leftrightarrow \neg\neg \exists z (z \,\underline{\mathtt{rn}}\, A) \wedge x{\downarrow}$.
The following lemmas are easily proved by induction on A.

LEMMA. *(Definedness of realizing terms; Substitution Property) For* $\mathrm{R} \in \{\underline{\mathtt{rn}}, \underline{\mathtt{rnt}}\}$

(i) $\vdash t \,\mathrm{R}\, A \to t{\downarrow}$,

(ii) $(x \,\mathrm{R}\, A)[y/t] \equiv x \,\mathrm{R}\, (A[y/t])$ $(x \notin \mathrm{FV}(A) \cup \mathrm{FV}(t), y \not\equiv x)$.

Proof. By induction on the complexity of A. Let e.g. $t \,\underline{\mathtt{rn}}\, \exists y A$, then $\mathbf{p}_1 t \,\underline{\mathtt{rn}}\, A[y/\mathbf{p}_0 t]$, hence by induction hypothesis $\mathbf{p}_1 t{\downarrow}$, and so by strictness $t{\downarrow}$. □

LEMMA. $\mathbf{HA}^* \vdash t \,\underline{\mathtt{rnt}}\, A \to A$.
A similar lemma holds for all combinations of realizability with truth (i.e. realizabilities with $\underline{\mathtt{t}}$ in their mnemonic code) we shall encounter in the sequel; we shall not bother to state it explicitly in the future. We can readily prove that realizability is sound for $\mathbf{HA}^*$:

1.7. THEOREM. *(Soundness theorem)*

$$\mathbf{HA}^* \vdash A \Rightarrow \mathbf{HA}^* \vdash t \,\underline{\mathtt{rn}}\, A \wedge t \,\underline{\mathtt{rnt}}\, A$$

for a suitable term t with $\mathrm{FV}(t) \subset \mathrm{FV}(A)$.

Proof. The proof proceeds by induction on the length of derivations; that is to say, we have to find realizing terms for the axioms, and for the rules we must show how to find a realizing term for the conclusion from realizing terms for the premises. We check some cases.

L5. Assume $t \,\underline{\mathtt{rn}}\, (A \to B)$, $t' \,\underline{\mathtt{rn}}\, (A \to C)$, and $x \,\underline{\mathtt{rn}}\, A$; then $\mathbf{p}(t \bullet x, t' \bullet x) \,\underline{\mathtt{rn}}\, (B \wedge C)$, so $\Lambda x.\mathbf{p}(t \bullet x, t' \bullet x) \,\underline{\mathtt{rn}}\, (A \to B \wedge C)$.

L14. Assume $t \,\underline{\mathtt{rn}}\, (A \to B)$, $x \notin \mathrm{FV}(B)$, and let $y \,\underline{\mathtt{rn}}\, \exists x\, A$, then $\mathbf{p}_1 y \,\underline{\mathtt{rn}}\, A[x/\mathbf{p}_0 y]$, hence $t[x/\mathbf{p}_0 y] \bullet (\mathbf{p}_1 y) \,\underline{\mathtt{rn}}\, B$, so $\Lambda y.t[x/\mathbf{p}_0 y] \bullet (\mathbf{p}_1 y) \,\underline{\mathtt{rn}}\, (\exists x\, A \to B)$.

Of the non-logical axioms, only induction requires attention. Suppose

$$x \,\underline{\mathtt{rn}}\, (A[y/0] \wedge \forall y(A \to A[y/Sy])).$$

Then

$$\mathbf{p}_0 x \,\underline{\mathtt{rn}}\, A[y/0], \quad z \,\underline{\mathtt{rn}}\, A \to (\mathbf{p}_1 x) \bullet y \bullet z \,\underline{\mathtt{rn}}\, A[y/Sy].$$

So let t be such that

$$t \bullet 0 \simeq \mathbf{p}_0 x, \quad t \bullet (Sy) \simeq (\mathbf{p}_1 x) \bullet y \bullet (t \bullet y).$$

The existence of t follows either by an application of the recursion theorem, or is immediate if closure under recursion has been built directly into the definition of recursive function. It is now easy to prove by induction that t realizes induction for A. □

A statement weaker than soundness is $\vdash A \Rightarrow \vdash \exists x(x \,\underline{\mathtt{rn}}\, A)$; we might call this *weak soundness*. We can also prove a stronger version of soundness:

1.8. THEOREM. *(Strong Soundness Theorem) For closed A*

$$\mathbf{HA}^* \vdash A \Rightarrow \mathbf{HA}^* \vdash \bar{n} \,\underline{\mathtt{rn}}\, A \wedge \bar{n} \,\underline{\mathtt{rnt}}\, A \quad \textit{for some numeral } \bar{n}.$$

Proof. Let $\mathbf{HA}^* \vdash A$; from the soundness theorem we find a term t such that

$$t \,\underline{\mathtt{rn}}\, A, \text{ hence } t\downarrow.$$

$t\downarrow$, i.e. $t = t$ is equivalent to a Σ^0_1-formula of $\mathbf{HA}$, say $\exists x(s = 0)$, and $\mathbf{HA}$ proves only true Σ^0_1-formulas, from which we see that $t = \bar{n}$ must be provable in $\mathbf{HA}^*$ for some numeral $\bar{n}$. Similarly for $\underline{\mathtt{rnt}}$. □

1.9. Remark. If one formalizes the proof of the soundness theorem, it is easy to see that there are primitive recursive functions ψ, ϕ such that

$$\mathbf{HA} \vdash \mathrm{Prf}(x, \ulcorner A \urcorner) \to \mathrm{Prf}(\phi(x), \mathrm{Sub}(\ulcorner y \,\underline{\mathtt{rn}}\, A \urcorner, y, \psi(x)))$$

where "Prf" is the formalized proof-predicate of $\mathbf{HA}^*$, $\ulcorner \xi \urcorner$ is the gödelnumber of expression ξ, and $\mathrm{Sub}(\ulcorner B \urcorner, x, \ulcorner s \urcorner)$ is the gödelnumber of $B[x/s]$.

In fact, the whole implication is provable even in primitive recursive arithmetic. But the statement expressing a formalized version of the *strong* completeness theorem:

$$\mathrm{Prf}(x, \ulcorner A \urcorner) \to \mathrm{Prf}(\phi(x), \ulcorner \overline{\psi(x)} \,\underline{\mathtt{rn}}\, A \urcorner)$$

(A closed, for suitable provably recursive ϕ, ψ) is not provable in $\mathbf{HA}$ (see section 1.16).

1.10. Lemma. *(Self-realizing formulas) For $\exists$-free formulas, canonical realizers exist, that is to say for each $\exists$-free A we have in* $\mathbf{HA}^*$

(i) $\vdash \exists x(x \,\underline{\mathtt{rn}}\, A) \to A$,

(ii) $\vdash A \to t_A \,\underline{\mathtt{rn}}\, A$ *for some term* t_A *with* $\mathrm{FV}(t_A) \subset \mathrm{FV}(A)$.

(iii) A formula A is provably equivalent to its own realizability, i.e. $A \leftrightarrow \exists x(x \,\underline{\mathtt{rn}}\, A))$, iff A is provably equivalent to an existentially quantified $\exists$-free formula.

(iv) Realizability is idempotent, i.e. $\exists x(x \,\underline{\mathtt{rn}}\, \exists y(y \,\underline{\mathtt{rn}}\, A)) \leftrightarrow \exists x(x \,\underline{\mathtt{rn}}\, A)$; in fact,even $\exists x(x \,\underline{\mathtt{rn}}\, (A \leftrightarrow \exists y(y \,\underline{\mathtt{rn}}\, A)))$ holds.

Proof. Take $t_{s=s'} := 0$, $t_{A\wedge B} := \mathbf{p}(t_A, t_B)$, $t_{\forall x A} := \Lambda x.t_A$, $t_{A\to B} := \Lambda x.t_B$ ($x \notin \mathrm{FV}(t_B)$), and prove (i) and (ii) by simultaneous induction on A. (iii) and (iv) are immediate corollaries. □

Remark. An observation of practical usefulness is the following. For any definable predicate with canonical realizers (i.e. a predicate A definable by an $\exists$-free formula) we obtain an equivalent realizability if we read restricted quantifiers $\forall x(A(x) \to \dots)$ and $\exists x(A(x) \wedge \dots)$ as quantifiers $\forall x{\in}A$, $\exists x{\in}A$ over a new domain with realizability clauses copied from numerical quantification, i.e.

$$x \,\underline{\mathtt{rn}}\, \forall y{\in}A.B := \forall y{\in}A(x{\bullet}y \,\underline{\mathtt{rn}}\, B) \wedge x{\downarrow},$$
$$x \,\underline{\mathtt{rn}}\, \exists y{\in}A.B := \mathbf{p}_1 x \,\underline{\mathtt{rn}}\, B[x/\mathbf{p}_0 x] \wedge A(\mathbf{p}_0 x).$$

In short, we may simply forget about the canonical realizers.

1.11. *Axiomatizing provable realizability*

As we have seen already in the introduction, realizability validates more than what is provable in **HA**; in fact, we can formally prove realizability of in $\mathbf{HA}^*$ an intuitionistic version of Church'sChurch, A. thesis:

$$\mathrm{CT}_0 \qquad \forall x \exists y\, A(x,y) \to \exists z \forall x (A(x, z\bullet x) \wedge z\bullet x\downarrow).$$

CT_0 is certainly not *provable in* **HA**, since it is in fact refutable in classical arithmetic. This version of Church'sChurch, A. thesis is in fact a combination of the well-known version which states "Each humanly computable function is recursive" and the intuitionistic reading of $\forall x \exists y A(x,y)$ which states that there is a method for constructing, for each given x, a y such that $A(x,y)$. Such a method describes a humanly computable function.

We now ask ourselves: is there a reasonably simple axiomatization (by a few axiom schemata say) of the formulas provably realizable in **HA**? The answer is yes, the provably realizable formulas can be axiomatized by a generalization of CT_0, namely *"Extended Church's Thesis"*:Church, A.

$$\mathrm{ECT}_0 \qquad \forall x(Ax \to \exists y\, Bxy) \to \exists z \forall x(Ax \to z\bullet x\downarrow \wedge B(x, z\bullet x)) \quad (A\ \ \exists\text{-free}).$$

Lemma. *Each instance of* ECT_0 *is* $\mathbf{HA}^*$*-realizable.*
Proof. Suppose

$$u\, \underline{\mathtt{rn}}\, \forall x(Ax \to \exists y Bxy)$$

Then $\forall xv(v\, \underline{\mathtt{rn}}\, Ax \to u\bullet x\bullet v\, \underline{\mathtt{rn}}\, \exists y Bxy)$, and since A is $\exists$-free, in particular $\forall x(Ax \to u\bullet x\bullet t_A\, \underline{\mathtt{rn}}\, \exists y Bxy)$, so $\forall x(Ax \to \mathbf{p}_1(u\bullet x\bullet t_A)\, \underline{\mathtt{rn}}\, B(x, \mathbf{p}_0(u\bullet x\bullet t_A))$. Then it is straightforward to see that

$$\mathbf{p}(\Lambda x.\mathbf{p}_0(u\bullet x\bullet t_A), \Lambda xv.\mathbf{p}(0, \mathbf{p}_1(u\bullet x\bullet t_A)))$$

realizes the conclusion. □

Remark. The condition "A is $\exists$-free" in ECT_0 cannot be dropped: applying unrestricted ECT_0 to $Ax := \exists z Txxz \vee \neg\exists z Txxz$, $Bxy := (y = 0 \wedge \exists z Txxz) \vee (y = 1 \wedge \neg\exists z Txxz)$ yields a contradiction. In fact, this example can be used to show that even unrestricted $\mathrm{ECT}_0!$ fails ($\mathrm{ECT}_0!$ is like ECT_0 except that $\exists y$ in the premise is replaced by $\exists! y$; $\exists! y$ means "there is a unique y such that").

Theorem. *(Characterization Theorem for* $\underline{\mathtt{rn}}$*-realizability)*

(i) $\mathbf{HA}^* + \mathrm{ECT}_0 \vdash A \leftrightarrow \exists x(x\, \mathrm{R}\, A)$ *for* $\mathrm{R} \in \{\underline{\mathtt{rn}}, \underline{\mathtt{rnt}}\}$,

(ii) For closed A, $\mathbf{HA}^* + \mathrm{ECT}_0 \vdash A \Leftrightarrow \mathbf{HA}^* \vdash \bar{n}\, \underline{\mathtt{rn}}\, A$ *for some numeral* $\bar{n}$.

Proof. (i) is proved by a straightforward induction on A. The crucial case is $A \equiv B \to C$; then $B \to C \leftrightarrow (\exists x(x\,\underline{\mathtt{rn}}\,B) \to \exists y(y\,\underline{\mathtt{rn}}\,C))$ (by the induction hypothesis) $\leftrightarrow \forall x(x\,\underline{\mathtt{rn}}\,B \to \exists y(y\,\underline{\mathtt{rn}}\,C))$ (by pure logic) $\leftrightarrow \exists z \forall x(x\,\underline{\mathtt{rn}}\,B \to z{\bullet}x\,\underline{\mathtt{rn}}\,C)$ (by ECT_0, since $x\,\underline{\mathtt{rn}}\,B$ is $\exists$-free) $\equiv \exists z(z\,\underline{\mathtt{rn}}\,(B \to C))$.

(ii). The direction $\Rightarrow$ follows from the strong soundness theorem plus the lemma; $\Leftarrow$ is an immediate consequence of (i). □

Curiosity prompts us to ask which formulas are classically provably realizable, i.e. provably realizable in first-order Peano Arithmetic **PA**, which is just **HA** with classical logic. The answer is contained in the following

PROPOSITION. $\mathbf{PA} \vdash \exists x(x\,\underline{\mathtt{rn}}\,A) \;\Leftrightarrow\; \mathbf{HA} + \mathrm{M} + \mathrm{ECT}_0 \vdash \neg\neg A$, *where* M *is* Markov's principle*:*

$$\mathrm{M} \qquad \forall x(A \vee \neg A) \wedge \neg\neg\exists x\, A \to \exists x\, A.$$

Proof. Let $\mathbf{PA} \vdash \exists x(x\,\underline{\mathtt{rn}}\,A)$, and let B be a negative formula (i.e. a formula in the $\wedge, \forall, \to$-fragment) such that $\mathbf{HA} + \mathrm{M} \vdash x\,\underline{\mathtt{rn}}\,A \leftrightarrow B(x)$. Then $\mathbf{PA} \vdash \neg\forall x \neg(x\,\underline{\mathtt{rn}}\,A)$, and since **PA** is conservative over **HA** for negative formulas (in consequence of Gödel's negative translation), also $\mathbf{HA} \vdash \neg\forall x \neg B$, i.e. $\mathbf{HA} + \mathrm{M} \vdash \neg\neg\exists x(x\,\underline{\mathtt{rn}}\,A)$, and thus it follows that $\mathbf{HA} + \mathrm{M} + \mathrm{ECT}_0 \vdash \neg\neg A$. The converse is simpler. □

1.12. *Extensions of* $\mathbf{HA}^*$

For suitable sets Γ of extra axioms, we may replace $\mathbf{HA}^*$ in the soundness and characterization theorem by $\mathbf{HA}^* + \Gamma$. Weak soundness and the characterization theorem require for all $A \in \Gamma$

$$(1) \qquad \mathbf{HA}^* + \Gamma \vdash \exists x(x\,\underline{\mathtt{rn}}\,A).$$

Soundness requires for all $A \in \Gamma$

$$(2) \qquad \mathbf{HA}^* + \Gamma \vdash t\,\underline{\mathtt{rn}}\,A \text{ for some term } t,$$

and strong soundness requires (2) and in addition: $\mathbf{HA}^* + \Gamma$ proves only true Σ^0_1-formulas.

EXAMPLES

(a) For Γ any set of $\exists$-free formulas soundness and the characterization theorem extend. If $\mathbf{HA}^* + \Gamma$ proves only true Σ^0_1-formulas, strong soundness holds. The next two examples permit characterization and strong soundness.

(b) Let $\prec$ be a primitive recursive well-ordering of $\mathbb{N}$, provably total and linear in $\mathbf{HA}^*$; for Γ we take all instances of *transfinite induction over* $\prec$:

$$\mathrm{TI}(\prec) \qquad \forall y(\forall x{\prec}y\, A \to A[x/y]) \to \forall x A.$$

(c) Γ is the set of instances of Markov's principle (cf. the last proposition in 1.11). In fact, in the presence of CT_0, which is valid under realizability, Γ may be replaced by a single axiom:

$$\forall xy(\neg\neg\exists zTxyz \to \exists zTxyz).$$

It is also worth noting that in the presence of M, we can use the following variant of ECT_0 which is equivalent to ECT_0:

$$\mathrm{ECT}_0' \qquad \forall x(\neg A \to \exists yBxy) \to \exists z\forall x(\neg A \to z\bullet x\downarrow \wedge B(x, z\bullet y)).$$

(d) An extension of another kind is obtained if we enrich the language with constants for inductively defined predicates, e.g. the tree predicate Tr. Intuitively, Tr is the least set containing the (code of the) single-node tree (i.e. $\langle\ \rangle \in \mathrm{Tr}$), and with every recursive sequence of tree codes $n\bullet 0, n\bullet 1, \ldots, n\bullet m, \ldots$ in Tr, Tr also contains a code for the infinite tree having the trees with codes $n\bullet m$ as immediate subtrees, namely $\mathbf{p}(1, n)$. Thus if

$$A(X, x) := (x = 0) \vee (\mathbf{p}_0 x = 1 \wedge \forall m(\mathbf{p}_1 x\bullet m \in X))$$

we have

$$A(\mathrm{Tr}, x) \to x \in \mathrm{Tr},$$
$$\forall x(A(\lambda y.B, x) \to B[y/x]) \to \forall x \in \mathrm{Tr}.B[y/x]$$

for all B in the language extended with the new primitive predicate Tr. Then we can extend $\underline{\mathtt{rn}}$-realizability simply by putting

$$x\,\underline{\mathtt{rn}}\,(t \in \mathrm{Tr}) := t \in \mathrm{Tr}.$$

Let us check that the soundness theorem extends. $A(\mathrm{Tr}, x)$ is equivalent to an $\exists$-free formula, so its realizability implies its truth, and $x \in \mathrm{Tr}$ follows. As to the schema, assume

$$u\,\underline{\mathtt{rn}}\,\forall x(A(\lambda y.B, x) \to B[y/x]), \text{ or}$$
$$u\,\underline{\mathtt{rn}}\,\forall x[(x = 0 \to B(0)) \wedge (\mathbf{p}_0 x = 1 \wedge \forall yB(\mathbf{p}_1 x\bullet y) \to Bx)].$$

So

$$\mathbf{p}_0(u\bullet 0)\bullet(0, 0)\,\underline{\mathtt{rn}}\,B(0),$$
$$\mathbf{p}_1(u\bullet x)\bullet v\,\underline{\mathtt{rn}}\,B(x) \text{ if } \mathbf{p}_0 x = 1 \text{ and } v\,\underline{\mathtt{rn}}\,(\mathbf{p}_0 x = 1 \wedge \forall yB(\mathbf{p}_1 x\bullet y).$$

Assume $\forall y(e\bullet(\mathbf{p}_1 x\bullet y)\,\underline{\mathtt{rn}}\,B(\mathbf{p}_1 x\bullet y))$, $\mathbf{p}_0 x = 1$. Then

$$v = \mathbf{p}(0, \Lambda y.e\bullet(\mathbf{p}_1 x\bullet y))\,\underline{\mathtt{rn}}\,(\mathbf{p}_0 x = 1 \wedge \forall yB(\mathbf{p}_1 x\bullet y)).$$

Therefore

$$\text{if } \mathbf{p}_0 x = 1 \text{ and } \forall y(e\bullet(\mathbf{p}_1 x\bullet y)\,\underline{\mathtt{rn}}\,B(\mathbf{p}_1 x\bullet y))$$
$$\text{then } \mathbf{p}_1(u\bullet x)\bullet(0, \Lambda y.e\bullet(\mathbf{p}_1 x\bullet y))\,\underline{\mathtt{rn}}\,B(x).$$

Now we construct by the recursion theorem an e such that

$$e\bullet x \simeq \begin{cases} \mathbf{p}_0(u\bullet 0)\bullet 0 \text{ if } x = 0, \\ \mathbf{p}_1(u\bullet x)\bullet \mathbf{p}(0, \Lambda y.e\bullet(\mathbf{p}_1 x\bullet y)) \text{ if } \mathbf{p}_0 x = 1, \\ \text{undefined otherwise.} \end{cases}$$

We then prove by induction on Tr that $\forall x \in \mathrm{Tr}(e\bullet x \,\underline{\mathtt{rn}}\, B(x))$. This is straightforward. This example is capable of considerable generalization, namely to arithmetic enriched with constants for predicates introduced by iterated inductive definitions of higher level; see e.g. (**?**, IV, section 6).

The examples just mentioned also permit extension of $\underline{\mathtt{rnt}}$-realizability. We end the section with some applications of $\underline{\mathtt{rn}}$- and $\underline{\mathtt{rnt}}$-realizability.

1.13. PROPOSITION. *(Consistency and inconsistency results)*

(i) $\mathbf{HA}^* + \mathrm{ECT}_0$ *is consistent relative to* $\mathbf{HA}^*$ *(and hence also relative to* $\mathbf{PA}$*).*

(ii) $\neg\forall x(A \vee \neg A)$, $\neg(\forall x\neg\neg B \to \neg\neg\forall x B)$ *are consistent with* $\mathbf{HA}^*$ *for certain arithmetical* A, B.

(iii) The schema "Independence of Premise"

IP $\qquad (\neg A \to \exists z B) \to \exists z(\neg A \to B)$

is not derivable in $\mathbf{HA}^*+\mathrm{CT}_0+\mathrm{M}$*; in fact,* $\mathbf{HA}^*+\mathrm{IP}+\mathrm{CT}_0+\mathrm{M} \vdash 1 = 0$.

Proof. (i) Immediate from the characterization theorem.

(ii) is a corollary of the realizability of CT_0: take $A \equiv \exists y Txxy$, $B \equiv \exists y Txxy \vee \neg\exists y Txxy$.

(iii) By M, $\neg\neg\exists y Txxy \to \exists z Txxz$; apply IP to obtain $\forall x\exists z(\neg\neg\exists y Txxy \to Txxz)$, then by CT_0 there is a total recursive F such that $\neg\neg\exists y Txxy \to T(x, x, Fx)$, and this would make $\exists y Txxy$ recursive in x. □

We next give an example of a conservative extension result.

1.14. DEFINITION. $\mathrm{CC}(\underline{\mathtt{rn}})$ (the $\underline{\mathtt{rn}}$-Conservative Class) is the class of formulas A such that whenever $B \to C$ is a subformula of A, then B is $\exists$-free. □

LEMMA. *For* $A \in \mathrm{CC}(\underline{\mathtt{rn}})$ *we have* $\vdash \exists x(x \,\underline{\mathtt{rn}}\, A) \to A$.

Proof. By induction on the structure of A. Consider the case $A \equiv B \to C$; then B is $\exists$-free, so there is a t_B such that $\vdash B \to t_B \,\underline{\mathtt{rn}}\, B$. Assume B and $x \,\underline{\mathtt{rn}}\, (B \to C)$, then $x\bullet t_B\downarrow \;\wedge\; x\bullet t_B \,\underline{\mathtt{rn}}\, C$, hence by the induction hypothesis C; therefore $(x \,\underline{\mathtt{rn}}\, (B \to C)) \to (B \to C)$. □

The lemma in combination with the characterization theorem yields

Proposition. $\mathbf{HA}^* + \mathrm{ECT}_0$ *is conservative over* $\mathbf{HA}^*$ *w.r.t. formulas in* $\mathrm{CC}(\underline{\mathtt{rn}})$*:*

$$(\mathbf{HA}^* + \mathrm{ECT}_0) \cap \mathrm{CC}(\underline{\mathtt{rn}}) = \mathbf{HA}^* \cap \mathrm{CC}(\underline{\mathtt{rn}}).$$

The following proposition follows from $\underline{\mathtt{rnt}}$-realizability.

1.15. Proposition. *(Derived rules) In* $\mathbf{HA}^*$

(i) *For sentences* $\vdash A \vee B \Rightarrow \vdash A$ *or* $\vdash B$ *(Disjunction property* DP*),*

(ii) *For sentences* $\vdash \exists x A \quad \Rightarrow \vdash A[x/\bar{n}]$ *for some numeral* $\bar{n}$ *(Explicit Definability for Numbers* EDN*),*

(iii) *Extended Church'sChurch, A. Rule: for* $\exists$*-free* A

$$\text{ECR} \qquad \vdash \forall x(A \to \exists y Bxy) \;\Rightarrow\; \vdash \exists z \forall x(A \to z{\bullet}x{\downarrow} \wedge B(x, z{\bullet}x)).$$

Proof. (i) follows from (ii) (actually, (i) and (ii) are equivalent for systems containing a minimum of arithmetic, see (**?**)). As to (ii), let $\vdash \exists x A$, then by the strong soundness for $\underline{\mathtt{rnt}}$-realizability $\vdash \bar{m}\, \underline{\mathtt{rnt}}\, \exists x A$ for some numeral $\bar{m}$, so $\vdash \mathbf{p}_1\bar{m}\, \underline{\mathtt{rnt}}\, A[x/\mathbf{p}_0\bar{m}]$, and hence $\vdash A[x/\mathbf{p}_0\bar{m}]$.

(iii) Assume $\vdash \forall x(A \to \exists y Bxy)$, then for a suitable t $\vdash t\, \underline{\mathtt{rnt}}\, \forall x(A \to \exists y Bxy)$, i.e.

$$\vdash \forall x \forall z(z\, \underline{\mathtt{rnt}}\, A \to \mathbf{p}_1(t{\bullet}x{\bullet}z)\, \underline{\mathtt{rnt}}\, B(x, \mathbf{p}_0(t{\bullet}x{\bullet}z)).$$

Since $t_A\, \underline{\mathtt{rnt}}\, A$,

$$\vdash \forall x(A \to \mathbf{p}_1(t{\bullet}x{\bullet}t_A)\, \underline{\mathtt{rnt}}\, B(x, \mathbf{p}_0(t{\bullet}x{\bullet}t_A))),$$

and therefore $\vdash \forall x(A \to B(x, \mathbf{p}_0(t{\bullet}x{\bullet}t_A))$. So we can take $z = \Lambda x.\mathbf{p}_0(t{\bullet}x{\bullet}t_A)$. $\square$

1.16. Remark. The DP cannot be formalized in any consistent extension of **HA** itself ((**?**), (**?**)). We sketch Myhill'sMyhill, J. R. argument (the result of FriedmanFriedman, H. M. is even stronger). Assume that there is a provably recursive function f satisfying

$$\vdash \mathrm{Prf}(x, \ulcorner A \vee B \urcorner) \to ((fx = 0 \wedge \mathrm{Pr}(\ulcorner A \urcorner)) \vee ((fx = 1 \wedge \mathrm{Pr}(\ulcorner B \urcorner))).$$

So $f = \{\bar{p}\}$, and $\vdash \forall x \exists y T\bar{n}xy$. Let F enumerate all primitive recursive functions, i.e. $\lambda n.F(i, n)$ is the i-th primitive recursive function. Put

$$D(n) := \bar{p}{\bullet}F(n, n) \neq 0,$$

then $\vdash \forall n(Dn \vee \neg Dn)$ (i.e. $\mathrm{Prf}(\bar{k}, \ulcorner \forall n(Dn \vee \neg Dn) \urcorner)$ for a specific $\bar{k}$), from which we can find a particular primitive recursive $\lambda n.F(\bar{m}, n)$ such that $\vdash$

$\mathrm{Prf}(F(\bar{m},\bar{n}),\ulcorner D\bar{n}\vee\neg D\bar{n}\urcorner)$. Then $D\bar{m} \to \bar{p}\bullet F(\bar{m},\bar{m}) \neq 0 \to \mathrm{Prf}(F(\bar{m},\bar{m}),\ulcorner D\bar{m}\vee \neg D\bar{m}\urcorner) \wedge \mathrm{Pr}(\ulcorner\neg D\bar{m}\urcorner)$, hence $\neg D\bar{m}$ follows, since $\mathbf{HA}^*$ is consistent. If we start assuming $\neg D\bar{m}$, we similarly obtain a contradiction.

From this we see that DP cannot be proved in $\mathbf{HA}^*$ itself; for if DP were provable in $\mathbf{HA}^*$, then a function f as above would be given by

$f(x) := \mathbf{p}_0$(the least y s.t.(x does not prove a closed disjunction and $y = 0$)
or (for some closed $\ulcorner A \vee B\urcorner$, $\mathrm{Prf}(x,\ulcorner A \vee B\urcorner) \wedge \mathbf{p}_0 y = 0 \wedge \mathrm{Prf}(\mathbf{p}_1 y,\ulcorner A\urcorner)$)
or (for some closed $\ulcorner A \vee B\urcorner$, $\mathrm{Prf}(x,\ulcorner A \vee B\urcorner) \wedge \mathbf{p}_1 y = 1 \wedge \mathrm{Prf}(\mathbf{p}_1 y,\ulcorner B\urcorner)$))).

This in turn implies that the strong soundness theorem is not formalizable in $\mathbf{HA}^*$, since strong soundness for $\underline{\underline{\mathtt{rn}}}$-realizability immediately implies EDN for $\mathbf{HA}^* + \mathrm{ECT}_0$.

1.17. *Notes*

Slash relations. Already in (**?**), a modification of numerical realizability was considered, namely $\Gamma \vdash$-realizability; let us use "R" as a short designation for this kind of realizability. The clauses for $\forall, \wedge$ and for prime formulas are as for ordinary realizability; the clauses for $\vee, \exists, \to$ become:

- $\mathbf{n}\,\mathrm{R}\,(A \vee B)$ iff $\mathbf{p}_0\mathbf{n} = 0$, $\mathbf{p}_1\mathbf{n}\,\mathrm{R}\,A$ and $\Gamma \vdash A$, or $\mathbf{p}_0\mathbf{n} \neq 0$, $\mathbf{p}_1\mathbf{n}\,\mathrm{R}\,B$ and $\Gamma \vdash B$;
- $\mathbf{n}\,\mathrm{R}\,\exists x A$ iff $\mathbf{p}_1\mathbf{n}\,\mathrm{R}\,A[x/\overline{\mathbf{p}_0\mathbf{n}}]$ and $\Gamma \vdash A[x/\overline{\mathbf{p}_0\mathbf{n}}]$;
- $\mathbf{n}\,\mathrm{R}\,(A \to B)$ iff for all $\mathbf{m}$, if $\mathbf{m}\,\mathrm{R}\,A$ and $\Gamma \vdash A$, then $\mathbf{n}\bullet\mathbf{m}$ is defined and $\mathbf{n}\bullet\mathbf{m}\,\mathrm{R}\,B$.

(**?**, Example 2 on page 510) used this notion to obtain a version of Church'sChurch, A. thesis. Later (**?**) observed that by dropping the realizability part and retaining only the provability part, one obtained an inductively defined property of formulas which could be used to obtain quite simple proofs of (generalizations of) the disjunction- and existence properties for logic and arithmetic. For easy reference, let us define $\Gamma|A$ ("Γ slashes A") for arithmetic, treating $\vee, \neg$ as defined, and putting $\Gamma|\vdash A$ as short for "$\Gamma|A$ and $\Gamma \vdash A$":

$$\begin{array}{ll}
\Gamma|P & \text{iff } \Gamma \vdash P \text{ for prime sentences } P,\\
\Gamma|(A \wedge B) & \text{iff } \Gamma|A \text{ and } \Gamma|B,\\
\Gamma|(A \to B) & \text{iff } \Gamma|\vdash A \Rightarrow \Gamma|B,\\
\Gamma|\exists x A & \text{iff } \Gamma|\vdash A[x/\bar{n}] \text{ for some numeral } \bar{n},\\
\Gamma|\forall x A & \text{iff } \Gamma|A[x/\bar{n}] \text{ for all numerals } \bar{n}.
\end{array}$$

$\Gamma|A$ for a formula A is defined as $\Gamma|B$ for some universal closure B of A. (For predicate logic, clauses for $\vee, \bot$ have to be added.)

"|" is sometimes called a realizability, but we think it better to reserve the term realizability for notions where realizing objects appear explicitly.

Since the "|" in "$\Gamma|A$" has nothing to do with division, the term "divides" for "|" is also not advisable. Therefore we call the notions derived from, or similar to Kleene'sKleene, S.C. $\Gamma|A$ simply *slash relations* or *slashes*.

In one respect $\Gamma|A$ is not well behaved; it is not closed under deduction, since it may happen that $\Gamma|A$, but not $\Gamma|(A \vee A)$. (**?**) gave a simple modification which overcomes this defect: the deducibility requirements in the clauses for $\vee, \exists$ are dropped, and for implication and universal quantification we require instead

$$\Gamma|(A \to B) \text{ iff } (\Gamma|A \Rightarrow \Gamma|B) \text{ and } \Gamma \vdash A \to B,$$
$$\Gamma|\forall x A x \text{ iff } \Gamma \vdash \forall x A x \text{ and } \Gamma|A[x/\bar{n}] \text{ for all } \bar{n}.$$

Now $\Gamma|A \Rightarrow \Gamma \vdash A$ holds for all A, and the modified slash yields the same applications as the original one. In fact, one easily proves by formula induction that $\Gamma| \vdash A$ in the sense of KleeneKleene, S.C. iff $\Gamma|A$ in the sense of Aczel. The Aczel slash also has an appealing model-theoretic interpretation; see e.g. (**?**, 13.7).

It is also worth noting that $C|C$ is both *necessary* and *sufficient* for the validity of the rule "For all A, $\vdash C \to \exists x A \Rightarrow \vdash C \to A[x/\bar{n}]$ for some $\bar{n}$" ((**?**), (**?**, 3.1.8)).

Slash operators in many variants have been widely used for obtaining metamathematical results for formalisms based on intuitionistic logic.

The slash as defined above applies to *sentences* only, but the use of partial reflection principles in combination with formalized versions of the slash relation, restricted to formulas of bounded complexity, may be used to deal with free numerical variables, see (**?**, 3.1.16).

Suitable slash relations for systems beyond arithmetic may be defined by considering conservative extensions with extra "witnessing constants" for existential statements. The explicit definability property for numbers EDN can then be proved by proving soundness of slash for the extended system (a typical example is (**?**)).

(**?**) describes the extension of the KleeneKleene, S.C. slash to higher-order logic. In (**?**) it is shown that this extension of the slash is in fact equivalent to a categorical construction on the free topos due to P. Freyd (see (**?**)).

(**?**) use slash relations *and* numerical realizability combined with truth ($\underline{\mathsf{q}}$-realizability, see below) to obtain the explicit set-existence property (explicit definability property for sets) for intuitionistic second-order arithmetic **HAS** (cf. 7.1) and intuitionistic set theory plus countable choice or relativized dependent choice.

(**?**) use a slash relation to establish a very interesting result: there is a particular number-theoretic property $A(n)$ such that if **HA** proves transfinite induction for a primitive recursive binary relation $\prec$ w.r.t. A, then $\prec$ is well-founded with ordinal less than ϵ_0. (The corresponding result is false for **PA**, cf. (**?**).) If transfinite induction is proved for $\prec$ w.r.t. A for the theory $\mathbf{HA}^+$ obtained by adding transfinite induction for all recursive-wellorderings, then $\prec$ is well-founded.

Of the many papers discussing or making use of slash relations we further mention: Beeson(?, ?, ?), (?), Dragalin(?, ?), (?), (?), (?), Myhill(?, ?), Robinson(?).

$\underline{\mathtt{q}}$-realizability. Since in soundness theorems for formalized realizability we prove deducibility instead of just truth, one can replace deducibility in the definition of $\Gamma\vdash$-realizability by truth; let us use "$\underline{\mathtt{q}}$" for this realizability. The clauses for $\exists, \to$ then become:

$$\begin{aligned} x\,\underline{\mathtt{q}}\,(A \to B) &:= \forall y(y\,\underline{\mathtt{q}}\,A \wedge A \to x{\bullet}y\,\underline{\mathtt{q}}\,B) \wedge x{\downarrow}, \\ x\,\underline{\mathtt{q}}\,\exists y A &:= \mathbf{p}_1 x\,\underline{\mathtt{q}}\,A[y/\mathbf{p}_0 x] \wedge A[y/\mathbf{p}_0 x]. \end{aligned}$$

Such a $\underline{\mathtt{q}}$-variant was used in (?) to obtain derived rules for intuitionistic analysis with function variables. $\underline{\mathtt{q}}$-realizability is also not closed under deducibility (think of an instance A of CT_0 unprovable in $\mathbf{HA}$; then A is $\underline{\mathtt{q}}$-realizable, but $A \vee A$ is not). (?) observed that an Aczel-style modification could be used instead of $\underline{\mathtt{q}}$-realizability; this corresponds to our $\underline{\mathtt{rnt}}$-realizability. (?) use $\underline{\mathtt{rn}}$-realizability and $\underline{\mathtt{q}}$-realizability to obtain consistency with Church'sChurch, A. thesis, the disjunction property and the numerical existence property for set theories based on intuitionistic logic, with axioms asserting the existence of very large cardinals, thereby demonstrating that the metamathematical properties just mentioned, often regarded as a test for the constructive character of a system, are not affected by assumptions concerning large cardinals.

Shanin's algorithm. In a number of papers ShaninShanin, N.A. presented a systematic way of making the constructive meaning of arithmetical formulas explicit. His method is logically equivalent to $\underline{\mathtt{rn}}$-realizability, as shown by (?). On the one hand Shanin'sShanin, N.A. algorithm is more complicated than realizability, on the other hand it has the advantage of being the identity on $\exists$-free formulas.

2 Abstract realizability and function realizability

2.1. After the leisurely introduction to numerical realizability in the preceding section, we now turn to variations and generalizations. In order to distinguish easily the various concepts of realizability, we shall use a certain mnemonic code:

$\underline{\mathtt{r}}$ signifies "realizability",
$\underline{\mathtt{n}}$ signifies "numerical" or "by numbers",
$\underline{\mathtt{f}}$ signifies "by functions",
$\underline{\mathtt{m}}$ signifies "modified",
$\underline{\mathtt{t}}$ signifies "combined with truth",
$\underline{\mathtt{l}}$ signifies "LifschitzLifschitz, V. variant of",
$\underline{\mathtt{e}}$ signifies "extensional".

Thus "$\underline{\mathtt{rft}}$" refers to "realizability by functions combined with truth" etc. Strictly speaking, the $\underline{\mathtt{r}}$ is redundant in many of these mnemonic codes.

A simple generalization of numerical realizability is realizability with a different set of realizing objects and/or different application operator; abstractly, the realizing objects with application have to form a combinatory algebra. We shall first sketch an abstract version of numerical realizability, namely realizability in a combinatory algebra with induction, then consider the interesting special case of function realizability.

2.2. Definition. (*The theory* **APP**) The language is single-sorted, based on LPT. The only non-logical predicate is N (natural numbers). There is an application operation $\bullet$ and constants

0 (zero), S (successor), P (predecessor),
$\mathbf{p}$, $\mathbf{p}_0$, $\mathbf{p}_1$ (pairing with inverses),
$\mathbf{k}$, $\mathbf{s}$ (combinators), $\mathbf{d}$ (numerical definition by cases).

(We have used the same symbols for pairing and inverses as in the case of $\mathbf{HA}^*$, even if there is a slight difference in syntax: $\mathbf{p}(t,t')$ in $\mathbf{HA}^*$ corresponds to $(\mathbf{p}t)t'$ in **APP**.) For $t_1{\bullet}t_2$ we simply write (t_1t_2), and we use association to the left, i.e. $t_1t_2\dots t_n$ is short for $(\dots((t_1t_2)t_3)\dots t_n)$.
Axioms for the constants:

$$\begin{aligned}
&N0,\ Nx \to N(Sx),\ Nx \to N(Px),\\
&P(St) \simeq t,\ P0 = 0,\ 0 \neq St,\\
&\mathbf{k}x\downarrow,\ \mathbf{k}ty \simeq t,\ \mathbf{s}xy\downarrow,\ \mathbf{s}tt't'' \simeq tt''(t't''),\\
&\mathbf{p}xy\downarrow,\ \mathbf{p}_0x\downarrow,\ \mathbf{p}_1x\downarrow,\ \mathbf{p}_0(\mathbf{p}tx) = t,\ \mathbf{p}_1(\mathbf{p}xt) = t,\\
&Nu \wedge Nv \to (u \neq v \to \mathbf{d}xyuv = x) \wedge \mathbf{d}xyuu = y.
\end{aligned}$$

Observe that by the general LPT-axioms we have $tt'\downarrow \to t\downarrow \wedge t'\downarrow$. Finally we have induction:

$$A[x/0] \wedge \forall x \in \mathrm{N}(A \to A[x/Sx]) \to \forall x \in \mathrm{N}.\, A \quad \Box$$

The combinators $\mathbf{k}, \mathbf{s}$ permit us to have λ-abstraction defined by induction on the construction of terms:

$$\begin{aligned}
&\lambda x.t := \mathbf{k}t \text{ for } t \text{ a constant or variable } \not\equiv x,\\
&\lambda x.x := \mathbf{skk},\\
&\lambda x.tt' := \mathbf{s}(\lambda x.t)(\lambda x.t').
\end{aligned}$$

For this definition

$$\begin{aligned}&\mathrm{FV}(\lambda x.t) \equiv \mathrm{FV}(t) \setminus \{x\},\\ &t'\downarrow \to (\lambda x.t)t' \simeq t[x/t'] \text{ if } t' \text{ is free for } x \text{ in } t,\\ &\lambda x.t\downarrow \text{ for all } t.\end{aligned}$$

It is not generally true that[2]

$$\text{if } x \notin \mathrm{FV}(t'), y \not\equiv x \text{ then } \lambda x.(t[y/t']) \simeq (\lambda x.t)[y/t'], \tag{1}$$

(consider e.g. $t \equiv y, t' \equiv \mathbf{kk}$) but we do have, for $x \notin \mathrm{FV}(t')$, $y \notin \mathrm{FV}(t'')$, $y \not\equiv x$

$$t''\downarrow \to ((\lambda x.t)[y/t'])t'' \simeq t[x/t''][y/t'] \equiv t[y/t'][x/t'']. \tag{2}$$

Property (1) can be guaranteed by an alternative definition of abstraction:

$$\begin{aligned}&\lambda' x.x := \mathbf{skk},\\ &\lambda' x.t := \mathbf{k}t \text{ if } x \notin \mathrm{FV}(t),\\ &\lambda' x.tt' := \mathbf{s}(\lambda' x.t)(\lambda' x.t') \text{ if } x \in \mathrm{FV}(tt'),\end{aligned}$$

but then we lose the property that $\lambda x.t\downarrow$ for all t. A recursor and a minimum operator may be defined with help of a fixed point operator (see e.g. (**?**, 9.3)) which permits us to define in **APP** all partial recursive functions. It follows that **HA** can be embedded into **APP** in a natural and straightforward way.

REMARK. Partial combinatory algebras are structures $(X, \bullet, \mathbf{k}, \mathbf{s})$, $\mathbf{k} \neq \mathbf{s}$, satisfying the relevant axioms above; in such structures we can always define terms forming a copy of $\mathbb{N}$, and appropriate $S, P, \mathbf{p}, \mathbf{p}_0, \mathbf{p}_1$, and we might simply have postulated induction for this particular copy of $\mathbb{N}$. However, in describing models it is more convenient not to be tied to a specific representation of $\mathbb{N}$ relative to the combinators. Also, we want to leave open the possibility that the interpretation of N is non-standard.

2.3. *The model of the partial recursive operations* PRO

The basic combinatory algebra is

$$(\mathbb{N}, \bullet, \Lambda xy.x, \Lambda xyz.x\bullet z\bullet(y\bullet z));$$

where $\bullet$ is partial recursive function application for $\mathbb{N}$. 0, S, P get their usual interpretation (more precisely, we choose codes $\Lambda x.Sx, \Lambda x.Px$ etc.; for $\mathbf{d}$ take $\Lambda uvxy[u \cdot \mathrm{sg}|x-y| + v(1 \dot{-} |x-y|)]$.

[2] This was overlooked in the proofs in (**?**, section 9.3), but is easily remedied by the use of (2). (**?**) recently showed that this problem might also be overcome by considering instead of **APP** based on combinators, a theory based on lambda-abstraction as a primitive (without a ξ-rule of the form $t = s \Rightarrow \lambda x.t = \lambda x.s$!) and an explicit substitution operator *as part of* the language.

In $\mathbf{HA}^*$ we can prove PRO to be a model of $\mathbf{APP}$, in the sense that $\mathbf{APP} \vdash A \Rightarrow \mathbf{HA}^* \vdash [\![A]\!]_{\mathrm{PRO}}$. Here and in the sequel we use "interpretation brackets": given some model $\mathcal{M}$, we use $[\![t]\!]_{\mathcal{M}}$, $[\![A]\!]_{\mathcal{M}}$ to indicate the interpretation of term t, formula A in the model $\mathcal{M}$. Thus "$[\![A]\!]_{\mathcal{M}}$ holds" means the same as $\mathcal{M} \models A$.

2.4. Definition. (*Abstract realizability*) $x \,\underline{\mathtt{r}}\, A$ in $\mathbf{APP}$ is defined by

$$\begin{array}{ll} x \,\underline{\mathtt{r}}\, P & := P \wedge x\downarrow \text{ for } P \text{ prime}, \\ x \,\underline{\mathtt{r}}\, (A \wedge B) & := (\mathbf{p}_0 x \,\underline{\mathtt{r}}\, A) \wedge (\mathbf{p}_1 x \,\underline{\mathtt{r}}\, B), \\ x \,\underline{\mathtt{r}}\, (A \to B) & := \forall y (y \,\underline{\mathtt{r}}\, A \to x \bullet y \,\underline{\mathtt{r}}\, B) \wedge x\downarrow, \\ x \,\underline{\mathtt{r}}\, \forall y\, A & := \forall y (x \bullet y \,\underline{\mathtt{r}}\, A), \\ x \,\underline{\mathtt{r}}\, \exists y\, A & := \mathbf{p}_1 x \,\underline{\mathtt{r}}\, A[y/\mathbf{p}_0 x]. \quad \Box \end{array}$$

Remark. $x \,\underline{\mathtt{r}}\, \forall y{\in}N.A$ becomes literally $\forall yz(z \,\underline{\mathtt{r}}\, (y \in N) \to (x \bullet y \bullet z) \,\underline{\mathtt{r}}\, A))$. It is easy to see that realizability with a special clause for the relativized quantifier

$$x \,\underline{\mathtt{r}}'\, \forall y{\in}N.A \;:=\; \forall y{\in}N (x \bullet y \,\underline{\mathtt{r}}'\, A)$$

is in fact equivalent.

The $\exists$-free formulas play the same role in $\mathbf{APP}$ as they do in $\mathbf{HA}^*$, i.e. $\exists$-free formulas have canonical realizing terms, their realizability coincides with their truth, and equivalence of realizability with truth for a formula A means that A is equivalent to a formula $\exists x B$, B $\exists$-free; the schema characterizing $\underline{\mathtt{r}}$-realizability is an *Extended Axiom of Choice*

EAC $\qquad \forall x(Ax \to \exists y Bxy) \to \exists z \forall x (Ax \to z \bullet x\downarrow \wedge B(x, z \bullet x))$ (A is $\exists$-free)

etc.

We may specialize $\underline{\mathtt{r}}$-realizability to PRO-$\underline{\mathtt{r}}$-realizability by interpreting $\mathbf{APP}$ in PRO. It is then not difficult to show that the resulting realizability of $\mathbf{HA}$ (as embedded in the obvious way into $\mathbf{APP}$) becomes equivalent to $\underline{\mathtt{rn}}$-realizability; cf. (**?**).

2.5. *Partial continuous function application*

Another important model of $\mathbf{APP}$ is the model PCO of functions with partial continuous application. Before we can discuss this, we need some preliminaries.

Notations. From primitive recursive $\mathbf{p}, \mathbf{p}_0, \mathbf{p}_1$ we can construct primitive recursive encodings $\mathbf{p}^n$ of n-tuples of natural numbers, with primitive recursive inverses $\mathbf{p}^n_i$ $(0 \le i < n)$. We may also assume finite sequences of natural numbers to be coded onto $\mathbb{N}$; we write $\langle n_0, \ldots, n_{x-1}\rangle$ for (the code of) the finite sequence $n_0, \ldots, n_{x-1}$; $\langle\ \rangle$ is the (code of the) empty sequence (which may be assumed to be equal to 0; see below).

If m is (a code of) a sequence, $\mathrm{lth}(m)$ is its length; $*$ is a primitive recursive concatenation function for codes of sequences. We abbreviate

$$\begin{array}{lll} n \preceq m & := & \exists n'(n * n' = m), \\ n \prec m & := & (n \preceq m \wedge n \neq m), \\ \hat{x} & := & \langle x \rangle. \end{array}$$

The primitive recursive inverse function $\lambda xy.(x)_y$ of sequence encoding satisfies

$$m = \langle n_0, \ldots, n_{x-1} \rangle \Rightarrow (m)_y = n_y \text{ for } y < x,\ (m)_y = 0 \text{ for } y \geq x.$$

For reasons of technical convenience we assume monotonicity in the arguments for encodings of pairs, n-tuples and finite sequences:

$$n < n' \rightarrow \mathbf{p}(n,m) < \mathbf{p}(n',m), \quad m < m' \rightarrow \mathbf{p}(n,m) < \mathbf{p}(n,m'),$$
and similarly for p-tuples;
$$n \leq n * m; \quad \mathrm{lth}(n) = \mathrm{lth}(m) \wedge \forall x((n)_x \leq (m)_x) \rightarrow n \leq m.$$

For example, for $\mathbf{p}$ we may take $\mathbf{p}(n,m) = \frac{1}{2}(n+m)(n+m+1)+m$. These monotonicity conditions in fact enforce $\langle\ \rangle = 0$. Encoding of n-tuples $(\alpha_1, \ldots, \alpha_n)$ of sequences is obtained by

$$(\alpha_0, \ldots, \alpha_n) := \lambda x.\mathbf{p}^n(\alpha_1 x, \ldots, \alpha_n x).$$

For initial segments of functions we use

$$\bar{\alpha}0 := \langle\ \rangle, \quad \bar{\alpha}(x+1) := \langle \alpha 0, \ldots, \alpha x \rangle. \quad \Box$$

DEFINITION. *Elementary Analysis* **EL** is a conservative extension of **HA** obtained by adding to **HA** variables $(\alpha, \beta, \gamma, \delta, \epsilon)$ and quantifiers for (total) functions from $\mathbb{N}$ to $\mathbb{N}$ (i.e. infinite sequences of natural numbers). There is λ-abstraction for explicit definition of functions, and a recursion-operator Rec such that (t a numerical term, ϕ a function term; $\phi(t,t') := \phi\mathbf{p}(t,t')$)

$$\mathrm{Rec}(t,\phi)(0) = t, \quad \mathrm{Rec}(t,\phi)(Sx) = \phi(x, \mathrm{Rec}(t,\phi)(x)).$$

Induction is extended to all formulas in the new language.

The functions of **EL** are assumed to be closed under "recursive in", which is expressed by including a weak choice axiom for quantifier-free A:

QF-AC $\quad \forall n \exists m A(n,m) \rightarrow \exists \alpha \forall n A(n, \alpha n) \quad \Box$

Definition. In **EL** we introduce abbreviations for *partial continuous application*

$$\begin{array}{rcl} \alpha(\beta) = x & := & \exists y(\alpha(\bar{\beta}y) = x + 1 \wedge \forall y'{<}y(\alpha(\bar{\beta}y') = 0), \\ \alpha|\beta = \gamma & := & \forall x(\lambda n.\alpha(\hat{x} * n)(\beta) = \gamma x) \wedge \alpha 0 = 0, \text{ or equivalently} \\ & & \forall x \exists y(\alpha(\hat{x} * \bar{\beta}y) = \gamma x + 1 \wedge \forall y'{<}y(\alpha(\hat{x} * \bar{\beta}y') = 0)) \wedge \alpha 0 = 0. \end{array}$$

We may introduce $|$, $\cdot(\cdot)$ as primitive operators in a conservative extension $\mathbf{EL}^*$ based on the logic of partial terms (1.3). □

Definition. $\mathbf{EL}^*$ is a conservative extension of **EL** based on the logic of partial terms, to which $\lambda\alpha\beta.\alpha|\beta$ and $\lambda\alpha\beta.\alpha(\beta)$ have been added as primitive operations. Numerical lambda-abstraction satisfies:

$$s{\downarrow} \wedge (\lambda x.t){\downarrow} \rightarrow (\lambda x.t)s = t[x/s], \quad (\lambda x.t){\downarrow} \leftrightarrow \forall x(t{\downarrow}).$$

For function application we require

$$\phi t{\downarrow} \leftrightarrow \phi{\downarrow} \wedge \phi{\downarrow}.$$

(The implication from left to right must hold since $\phi{\downarrow}$ is supposed to imply totality of the function denoted by ϕ.) For Rec we have

$$\text{Rec}(t, \phi){\downarrow} \leftrightarrow t{\downarrow} \wedge \phi{\downarrow}.$$

□

2.6. *The model of the partial continuous operations* PCO

This model has as domain all the total functions from $\mathbb{N}$ to $\mathbb{N}$; the application is $|$ defined above. The elementary theory of the model can be formalized in $\mathbf{EL}^*$. Some work is needed to show that PCO is actually a model of **APP**.

Definition. (*The class of neighbourhood functions*)

$$\alpha \in K^* \quad := \quad \alpha 0 = 0 \wedge \forall nm(\alpha n > 0 \rightarrow \alpha n = \alpha(n * m)) \wedge \forall \beta \exists x(\alpha(\bar{\beta}x) > 0).$$

□

Crucial is the following

Lemma. *To each function term ϕ, and each numerical term t of $\mathbf{EL}^*$, we can construct function terms $\Phi_\phi \in K^*$, $\Phi_t \in K^*$ respectively, such that*

(i) $\Phi_\phi|\alpha \simeq \phi$;

(ii) $(\Phi_t|\alpha){\downarrow}$ *iff* $t{\downarrow}$;

(iii) $t{\downarrow} \rightarrow (\Phi_t|\alpha)0 = t$;

(iv) $\mathrm{FV}(\Phi_t) \subset \mathrm{FV}(t) \setminus \{\alpha\}$, $\mathrm{FV}(\Phi_\phi) \subset \mathrm{FV}(\phi) \setminus \{\alpha\}$, Φ_t, Φ_ϕ *primitive recursive in their free variables.*

Proof. (i)–(iv) are proved by simultaneous induction on the construction of numerical and function terms. The reason that we need a function term with partial continuous application $|$ to represent a numerical term (instead of application $\cdot(\cdot)$) is that a numerical term t may contain function-terms as subterms, which all have to be defined by the strictness condition of the logic of partial terms; this is a Π^0_2-condition and cannot be expressed by definedness of a numerical term.

We consider a few typical cases. In all cases we put $\Phi_\phi 0 = 0$, $\Phi_t 0 = 0$.
Case 1. $t \equiv x$. Take $\Phi_t(\hat{z} * \bar{\alpha} n) = x + 1$. Similarly for $t \equiv 0$.
Case 2. $\phi \equiv \alpha$. Take

$$\Phi_\alpha(\hat{z} * \bar{\alpha} n) = \begin{cases} \alpha z + 1 \text{ if } z < n, \\ 0 \text{ otherwise.} \end{cases}$$

Case 3. $t \equiv \phi(\psi)$. Take

$$\Phi_{\phi(\psi)}(\hat{x} * \bar{\alpha} n) = \begin{cases} z + 1 \text{ if } \exists u{<}n \forall y{<}\mathrm{lth}(u)(\Phi_\psi(\hat{y} * \bar{\alpha} n) = (u)_y + 1 \wedge \\ \qquad \Phi_\phi(u) = z + 1) \wedge \Phi_\phi(\hat{x} * \bar{\alpha} n) > 0 \wedge \Phi_\psi(\hat{x} * \bar{\alpha} n) > 0, \\ 0 \text{ otherwise.} \end{cases}$$

Case 4. $\phi \equiv \psi | \xi$. Take

$$\Phi_{\psi|\xi}(\hat{x} * \bar{\alpha} n) = \begin{cases} z + 1 \text{ if } \exists u{<}n \forall y{<}\mathrm{lth}(u)(\Phi_\xi(\hat{y} * \bar{\alpha} n) = (u)_y + 1 \wedge \\ \qquad \Phi_\psi(\hat{x} * u) = z + 1) \wedge \Phi_\psi(\hat{x} * \bar{\alpha} n) > 0 \wedge \Phi_\xi(\hat{x} * \bar{\alpha} n) > 0, \\ 0 \text{ otherwise .} \end{cases}$$

The other cases are left to the reader. □

It is now easy to prove in $\mathbf{EL}^*$ that PCO is a model of **APP** (do not confuse the λ-abstraction in $\mathbf{EL}^*$ with the defined λ-operator in **APP**). For example, an interpretation of $[\![\mathbf{s}]\!]$ is found as follows. If ϕ is a function term of $\mathbf{EL}^*$, let us write $\Lambda\alpha.\phi$ for the Φ_ϕ given by the lemma. Then we put

$$[\![\mathbf{s}]\!]_{\mathrm{PCO}} := \Lambda\alpha\Lambda\beta\Lambda\gamma.(\alpha|\gamma)|(\beta|\gamma).$$

Clearly $([\![\mathbf{s}]\!]|\alpha)|\beta = \Lambda\gamma(\alpha|\gamma)|(\beta|\gamma)$, since all terms Φ_ϕ are total, hence defined. Moreover $(([\![\mathbf{s}]\!]|\alpha)|\beta)|\gamma \simeq (\alpha|\gamma)|(\beta|\gamma)$.

We spell out a definition of realizability for this application which is not literally what one obtains by interpreting $\underline{\mathbf{r}}$-realizability in PCO, but equivalent to it:

2.7. DEFINITION. *(Realizability by functions)* With each formula A of $\mathbf{EL}^*$ we associate $\alpha\,\underline{\mathtt{rf}}\,A$ ($\alpha \notin \mathrm{FV}(A)$) as follows:

$$\begin{array}{ll} \alpha\,\underline{\mathtt{rf}}\,P & := P \wedge \alpha{\downarrow}\ \ (P\ \text{prime}), \\ \alpha\,\underline{\mathtt{rf}}\,(A \wedge B) & := (\mathbf{p}_0\alpha\,\underline{\mathtt{rf}}\,A) \wedge (\mathbf{p}_1\alpha\,\underline{\mathtt{rf}}\,B), \\ \alpha\,\underline{\mathtt{rf}}\,(A \to B) & := \forall\beta(\beta\,\underline{\mathtt{rf}}\,A \to \alpha|\beta\,\underline{\mathtt{rf}}\,B) \wedge \alpha{\downarrow}, \\ \alpha\,\underline{\mathtt{rf}}\,\forall x\,A & := \forall x(\alpha|\lambda n.x\,\underline{\mathtt{rf}}\,A), \\ \alpha\,\underline{\mathtt{rf}}\,\forall\beta\,A & := \forall\beta(\alpha|\beta\,\underline{\mathtt{rf}}\,A), \\ \alpha\,\underline{\mathtt{rf}}\,\exists x\,A & := \mathbf{p}_1\alpha\,\underline{\mathtt{rf}}\,A[x/(\mathbf{p}_0\alpha)0], \\ \alpha\,\underline{\mathtt{rf}}\,\exists\beta\,A & := \mathbf{p}_1\alpha\,\underline{\mathtt{rf}}\,A[\beta/\mathbf{p}_0\alpha]. \end{array}$$

$\underline{\mathtt{rft}}$-realizability is defined by modifying $\underline{\mathtt{rf}}$-realizability as before. □

Now the theory runs to a large extent parallel to numerical realizability. The role of ECT_0 is taken over by the following schema of *Generalized Continuity*:

GC $\qquad \forall\alpha(A \to \exists\beta B(\alpha,\beta)) \to \exists\gamma\forall\alpha(A \to \gamma|\alpha{\downarrow} \wedge B(\alpha,\gamma|\alpha))$ (A $\exists$-free)

where $\exists$-*free* in $\mathbf{EL}^*$ is defined as before; in $\mathbf{EL}$, $\exists$-free formulas correspond to the class of formulas constructed from $t = s$, $\exists x(t = s)$, $\exists\alpha(t = s)$ by means of $\to, \wedge, \forall$.

2.8. PROPOSITION. *(Examples of applications) For* $\mathbf{EL}^*$ *we have*

(i) $\vdash \forall\alpha(A \to \exists\beta B(\alpha,\beta)) \ \Rightarrow \vdash \exists\gamma\forall\alpha(A \to \gamma|\alpha{\downarrow} \wedge B(\alpha,\gamma|\alpha))$ *for almost negative A (Generalized Continuity Rule* GCR*).*

(ii) For $\exists\alpha A\alpha$ closed, $\vdash \exists\alpha\,A\alpha \ \Rightarrow$ there exists some $\bar{n}$ such that $\vdash A(\{\bar{n}\}) \wedge \forall m(\bar{n}\bullet m{\downarrow})$, i.e. if $\vdash \exists\alpha\,A(\alpha)$, there is a total recursive function f such that $\vdash A(f)$.

(iii) $\mathrm{CC}(\underline{\mathtt{rf}}) \cap \mathbf{EL}^* = \mathrm{CC}(\underline{\mathtt{rf}}) \cap (\mathbf{EL}^* + \mathrm{GC})$*, where the conservative class* $\mathrm{CC}(\underline{\mathtt{rf}})$ *for $\underline{\mathtt{rf}}$-realizability is defined in complete analogy to* $\mathrm{CC}(\underline{\mathtt{rn}})$*.*

Proof. (Of (ii).) The strong soundness theorem yields in this case a particular function term ϕ such that $\vdash \phi\,\underline{\mathtt{rft}}\,\exists\alpha\,A\alpha$, hence $\vdash \mathbf{p}_1\phi\,\underline{\mathtt{rft}}\,A[\alpha/\mathbf{p}_0\phi]$, and thus $\vdash A[\alpha/\mathbf{p}_0\phi] \wedge \mathbf{p}_0\phi{\downarrow}$; $\mathbf{p}_0(\phi)$ is a closed function term in the language of $\mathbf{EL}$ which may be written as $\{\bar{n}\}$. ($\{\bar{n}\}$ is short for $\lambda x.(\bar{n}\bullet x)$.) □

2.9. *Examples of extensions*

Bar Induction for Decidable predicates is an induction principle

$$\mathrm{BI_D} \qquad \forall\alpha\exists x\,P(\bar{\alpha}x) \wedge \forall n(Pn \vee \neg Pn) \wedge \forall n(Pn \to Qn) \wedge \forall n(\forall m(Q(n * \langle m\rangle) \to Qn) \to Q\langle\,\rangle$$

An equivalent principle is BI! with $\forall\alpha\exists x\,P(\bar{\alpha}x) \wedge \forall n(Pn \vee \neg Pn)$ replaced by $\forall\alpha\exists! x\,P(\bar{\alpha}x)$. $\mathrm{BI_D}$ implies the *Fan theorem for Decidable predicates*

$$\mathrm{FAN_D} \qquad \forall\alpha{\le}\beta\exists x\,A(\bar{\alpha}x) \wedge \forall n(An \vee \neg An) \to \exists z\forall\alpha{\le}\beta\exists x{\le}z\,A(\bar{\alpha}x)$$

where $\alpha \leq \beta := \forall x(\alpha x \leq \beta x)$. Since $\underline{\texttt{rf}}$-realizability validates continuity principles, in fact the stronger

FAN $\qquad \forall\alpha\exists x\, A(\alpha, x) \rightarrow \exists z\forall\alpha{\leq}\beta\exists x{\leq}z\, A(\alpha, x)$

holds.

In the soundness and characterization theorems $\mathbf{EL}^*$ may be replaced by $\mathbf{EL}^* + \Gamma$, where for example Γ can be the set of all instances of one or more of the following schemata: M, TI($\prec$), $\mathrm{FAN_D}$, $\mathrm{BI_D}$.

2.10. *Notes*

(**?**) contains the first formalization of $\underline{\texttt{rf}}$-realizability. In this paper KleeneKleene, S.C. shows a.o. that formulas such that all their subformulas in the scope of an universal function quantifier are $\exists$-free are true iff $\underline{\texttt{rf}}$-realizable (provable classically). In (**?**) a thorough formalized treatment of function realizability is given, also of a (version of) $\underline{\texttt{rft}}$-realizability. Another notion of realizability by functions is found in (**?**, **?**).

(**?**) considers a notion of function-realizability which combines the idea of $\underline{\texttt{rf}}$-realizability with the topological model of elementary intuitionistic analysis in (**?**) (itself an adaptation of a model due to (**?**)).

(**?**) used an abstract version of realizability to show consistency of an axiom of choice with combinatory logic. Staples(**?**, **?**) used realizability with combinators for higher-order logic and set theory. Abstract realizability for theories including **APP** was introduced by Feferman(**?**, **?**).

Of the researches using abstract versions of realizability we further mention Beeson(**?**, **?**, **?**, **?**), Renardel de Lavalette(**?**, **?**).

3 Modified realizability

In the case of numerical and function realizability, we started with the concrete and ended with the abstract version.

For modified realizability on the other hand, it is advantageous to start with the abstract setting, and afterwards to specialize to more concrete versions. The abstract setting of modified realizability is not a type-free theory such as **APP**, sketched above, but a system $\mathbf{HA}^\omega$ of intuitionistic finite-type arithmetic.

3.1. *Description of intuitionistic finite-type arithmetic* $\mathbf{HA}^\omega$

The *set of finite type symbols* $\mathcal{T}$ is generated by the clauses $0 \in \mathcal{T}$ (type of the natural numbers); if $\sigma, \tau \in \mathcal{T}$ then $(\sigma \times \tau) \in \mathcal{T}$ (formation of product types) and $(\sigma \rightarrow \tau) \in \mathcal{T}$ (formation of *function types*). We use $\sigma, \sigma', \ldots, \tau, \tau', \ldots, \rho, \rho', \ldots$ for arbitrary type symbols.

As an alternative for $(\sigma \to \tau)$ we write $(\sigma\tau)$; 1 is short for (00), $n+1$ for $(n0)$. Outer parentheses in type symbols are usually omitted. Further saving on parentheses is obtained by the convention of association to the *right*, i.e. $\sigma_0\sigma_1\sigma_2\sigma_3$ abbreviates $(\sigma_0(\sigma_1(\sigma_2\sigma_3)))$; $\sigma_1 \times \sigma_2 \times \cdots \times \sigma_n$ abbreviates $(\cdots((\sigma_1 \times \sigma_2) \times \sigma_3) \cdots \times \sigma_n)$.

The language of of *intuitionistic finite-type arithmetic* $\mathbf{HA}^\omega$ is a many-sorted language with variables $(x^\sigma, y^\sigma, z^\sigma, \ldots)$ of all types; for each $\sigma \in \mathcal{T}$ there is a primitive equality $=_\sigma$, and there are some constants listed below, and an application operation $\mathrm{App}_{\sigma,\tau}$ from $\sigma \to \tau$ and σ to τ. For arbitrary terms we use $t, t', t'', \ldots, s, s', s'', \ldots$. In order to indicate that t is a term of type σ we write $t \in \sigma$ or t^σ. If $t \in \sigma \to \tau$, $t' \in \sigma$, then $\mathrm{App}_{\sigma,\tau}(t, t') \in \tau$. For $\mathrm{App}_{\sigma,\tau}(t, t')$ we simply write (tt') or even tt'; we save on parentheses by association to the *left*: $t_1 \ldots t_n$ is short for $(\cdots((t_1t_2)t_3)\cdots t_n)$. As constants we have for all $\sigma, \tau, \rho \in \mathcal{T}$:

$$
\begin{array}{l}
0 \in 0 \text{ (zero)}, \quad S \in 00 \text{ (successor)}, \\
\mathbf{p}^{\sigma,\tau} \in \sigma\tau(\sigma \times \tau), \text{ (pairing)} \\
\mathbf{p}_0^{\sigma,\tau} \in (\sigma \times \tau)\sigma, \ \ \mathbf{p}_1^{\sigma,\tau} \in (\sigma \times \tau)\tau, \text{ (unpairing)} \\
\mathbf{k}^{\sigma,\tau} \in \sigma\tau\sigma, \ \ \mathbf{s}^{\rho,\sigma,\tau} \in (\rho\sigma\tau)(\rho\sigma)\rho\tau \text{ (combinators)}, \\
\mathbf{r}^\sigma \in \sigma(\sigma 0 \sigma)0\sigma \text{ (recursor)}.
\end{array}
$$

Here again we use the same symbols ($\mathbf{k}, \mathbf{s}$, $\mathbf{p}, \mathbf{p}_0, \mathbf{p}_1$) for operations closely analogous to the operations denoted by the same symbols in **APP**. We shall drop type sub- en superscripts wherever it is safe to do so; types are always assumed to be "fitting" (i.e. if tt' is written then $t \in \sigma\tau$, $t' \in \sigma$ for suitable σ, τ).

The logical basis of $\mathbf{HA}^\omega$ is many-sorted intuitionistic predicate logic with equality; the constants satisfy the following equations:

$$
\begin{array}{l}
\mathbf{p}_0(\mathbf{p}xy) = x, \ \ \mathbf{p}_1(\mathbf{p}xy) = y, \ \ \mathbf{p}(\mathbf{p}_0x)(\mathbf{p}_1x) = x, \\
\mathbf{k}xy = x, \ \ \mathbf{s}xyz = xz(yz), \ \ \mathbf{r}xy0 = x, \ \ \mathbf{r}xy(Sz) = y(\mathbf{r}xyz)z.
\end{array}
$$

Finally, we have $0 \neq Sx$, $Sx = Sy \to x = y$, and full induction. (Actually, $Sx = Sy \to x = y$ is redundant, since we can define a predecessor function P such that $P(St) = x$.)

There is defined λ-abstraction, as in for **APP**; we can use the second recipe mentioned in 2.2, with $\lambda x.t = \mathbf{k}t$ for all t not containing x. **HA** is embedded in $\mathbf{HA}^\omega$ in the obvious way.

3.2. *The systems* $\mathbf{I}$-$\mathbf{HA}^\omega$, $\mathbf{E}$-$\mathbf{HA}^\omega$

$\mathbf{I}$-$\mathbf{HA}^\omega$, *intensional* finite-type arithmetic is a strengthening of $\mathbf{HA}^\omega$ obtained by including an equality functional $\mathbf{e}_\sigma \in \sigma\sigma 0$ for all $\sigma \in \mathcal{T}$, satisfying

$$
\mathbf{e}x^\sigma y^\sigma \leq 1, \ \ \mathbf{e}x^\sigma y^\sigma = 0 \leftrightarrow x^\sigma = y^\sigma,
$$

so equality is decidable at all types.

On the other hand, *extensional* finite-type arithmetic $\mathbf{E\text{-}HA}^\omega$ is obtained from $\mathbf{HA}^\omega$ by adding extensionality axioms for all types σ:

$$\forall x^\sigma (y^{\sigma\tau} x =_\tau z^{\sigma\tau} x) \leftrightarrow y^{\sigma\tau} =_{\sigma\tau} z^{\sigma\tau}.$$

This permits us to *define* equality of type σ in terms of $=_0$, via

$$y =_{\sigma\tau} z := \forall x^\sigma (yx =_\tau zx),$$
$$y =_{\sigma\times\tau} z := \mathbf{p}_0 y =_\sigma \mathbf{p}_0 z \wedge \mathbf{p}_1 y =_\tau \mathbf{p}_1 z.$$

Therefore we may assume $\mathbf{E\text{-}HA}^\omega$ to be formulated in a language which contains only $=_0$ as primitive equality, so that prime formulas are always decidable.

3.3. *Models of* $\mathbf{HA}^\omega$

A model of $\mathbf{HA}^\omega$ is given by a type structure $\langle M_\sigma, \sim_\sigma\rangle_{\sigma\in\mathcal{T}}$, with M_σ a set, $\sim_\sigma$ an equivalence relation on M_σ, plus suitable interpretations of $\mathrm{App}_{\sigma,\tau}$ and the various constants.

(i) FTS, the *Full Type Structure.* Take $\mathbb{N}$ for M_0, for $M_{\sigma\tau}$ take the set of all functions from M_σ to M_τ, for $M_{\sigma\times\tau}$ take $M_\sigma \times M_\tau$; this is the full type structure; $\sim_\sigma$ at each type is set-theoretic equality, and it is obvious how to interpret App and the constants.

(ii) HRO, the *Hereditarily Recursive Operations.* Put

$$\begin{aligned}
\mathrm{HRO}_0 &:= \mathbb{N},\\
\mathrm{HRO}_{\sigma\times\tau} &:= \{z : \mathbf{p}_0 z \in \mathrm{HRO}_\sigma \wedge \mathbf{p}_1 z \in \mathrm{HRO}_\tau\},\\
\mathrm{HRO}_{\sigma\tau} &:= \{z : \forall x \in \mathrm{HRO}_\sigma (z\bullet x \in \mathrm{HRO}_\tau)\}.
\end{aligned}$$

App is interpreted as partial recursive application (i.e. as $\bullet$), $=_\sigma$ as equality between numbers (as elements of HRO_σ),

$$[\![0]\!] := 0,\ [\![S]\!] := \Lambda x.Sx,\ [\![\mathbf{k}]\!] := \Lambda xy.x,\ [\![\mathbf{s}]\!] := \Lambda xyz.xz(yz),$$
$$[\![\mathbf{p}]\!] := \Lambda xy.\mathbf{p}(x,y),\ [\![\mathbf{p}_0]\!] := \Lambda x.\mathbf{p}_0 x,\ [\![\mathbf{p}_1]\!] := \Lambda x.\mathbf{p}_1 x,$$
$$[\![\mathbf{r}]\!] := \text{ a suitable code for a recursor},\ [\![\mathbf{e}]\!] := \Lambda xy.\mathrm{sg}|x-y|.$$

The existence of a suitable code for a recursor either follows directly from the definition of recursive function, or by an application of the recursion theorem yielding a solution r to $r\bullet(x,y,0) \simeq 0$, $r\bullet(x,y,Sz) \simeq y\bullet(r\bullet(x,y,z),z)$, as in (**?**, 3.7.5). The result is a model of $\mathbf{I\text{-}HA}^\omega$.

(iii) HEO, the model of the *Hereditarily Effective Operations.* We define a partial equivalence relation $\sim_\sigma$ between natural numbers for each $\sigma \in \mathcal{T}$ by

$$\begin{aligned}
x \sim_0 y &:= x = y,\\
x \sim_{\sigma\times\tau} y &:= (\mathbf{p}_0 x \sim_\sigma \mathbf{p}_0 y) \wedge (\mathbf{p}_1 x \sim_\tau \mathbf{p}_1 y),\\
x \sim_{\sigma\tau} y &:= \forall zz'(z \sim_\sigma z' \to x\bullet z \sim_\tau y\bullet z' \wedge x\bullet z \sim_\tau x\bullet z' \wedge y\bullet z \sim_\tau y\bullet z')
\end{aligned}$$

where

$$z \in \mathrm{HEO}_\sigma := z \sim_\sigma z.$$

For the rest, the definition of interpretations of $0, S, \mathbf{k}, \mathbf{s}$, $\mathbf{p}, \mathbf{p}_0, \mathbf{p}_1$, $\mathbf{r}$ proceeds as before, we interpret $=_\sigma$ as $\sim_\sigma$, and we obtain a model of $\mathbf{E\text{-}HA}^\omega$.

3.4. Definition. (*Modified realizability*) We define $x^\sigma \,\underline{\mathtt{mr}}\, A$, for formulas of $\mathbf{HA}^\omega$, by induction on the complexity of A as follows. The type σ of x is determined by the structure of A.

$$\begin{array}{ll} x^0 \,\underline{\mathtt{mr}}\, (t = s) & := (t = s), \\ x \,\underline{\mathtt{mr}}\, (A \wedge B) & := \mathbf{p}_0 x \,\underline{\mathtt{mr}}\, A \wedge \mathbf{p}_1 x \,\underline{\mathtt{mr}}\, B, \\ x \,\underline{\mathtt{mr}}\, (A \to B) & := \forall y (y \,\underline{\mathtt{mr}}\, A \to xy \,\underline{\mathtt{mr}}\, B), \\ x \,\underline{\mathtt{mr}}\, \forall x\, A & := \forall z (xz \,\underline{\mathtt{mr}}\, A), \\ x \,\underline{\mathtt{mr}}\, \exists z\, A & := \mathbf{p}_1 x \,\underline{\mathtt{mr}}\, A[z/\mathbf{p}_0 x]. \end{array}$$

We also consider $\underline{\mathtt{mrt}}$-realizability, which is similar to $\underline{\mathtt{rnt}}$-realizability. All clauses are the same as for $\underline{\mathtt{mr}}$, the implication clause excepted, which now reads

$$x \,\underline{\mathtt{mrt}}\, (A \to B) \;:=\; \forall y (y \,\underline{\mathtt{mrt}}\, A \to xy \,\underline{\mathtt{mrt}}\, B) \wedge (A \to B). \quad \Box$$

Remark. In the usual definition (cf. (**?**, 3.4.2)), one realizes with sequences of terms $\vec{t}$, of length and types depending on the structure of A. The attractive feature of this definition is that $\exists$-free formulas are literally self-realizing: for $\exists$-free A, $\vec{t} \,\underline{\mathtt{mr}}\, A \;:=\; A$, so $\vec{t}$ is empty.

For our definition above, the choice of type 0 for the realizing objects of prime formulas is somewhat arbitrary; a more canonical choice might have been obtained by (conservatively) adding a singleton type to $\mathbf{HA}^\omega$ and letting the single element of this type realize $t = s$ iff true.

A concrete version of $\underline{\mathtt{mr}}$-realizability is obtained by interpreting $\mathbf{HA}^\omega$ in a model $\mathcal{M}$; this yields $\mathcal{M}$-$\underline{\mathtt{mr}}$-realizability. The difference between Kleene'sKleene, S.C. realizability and $\underline{\mathtt{mr}}$-realizability becomes clear by comparing $\underline{\mathtt{rn}}$-realizability and HRO-$\underline{\mathtt{mr}}$-modified realizability of statements of the form

$$\forall y \neg \forall z \neg Txyz \to B$$

For $\underline{\mathtt{rn}}$, this requires a realizer t which must be applicable to the canonical realizer $\Lambda y.0$ of $\forall y \neg \forall z \neg Txyz$ *if* this is true. On the other hand, in the case of HRO-$\underline{\mathtt{mr}}$-realizability $t \bullet \Lambda y.0$ must be defined, whether $\forall y \neg \forall z \neg Txyz$ is true or not. In other words, in modified realizability, realizing objects for implications have a larger domain of definition than what is required by "pure" realizability.

Soundness now takes the form

3.5. THEOREM. *(Soundness)*

$$\mathbf{HA}^\omega \vdash A \;\Rightarrow\; \mathbf{HA}^\omega \vdash t\,\underline{\mathtt{mr}}\,A \wedge t\,\underline{\mathtt{mrt}}\,A \text{ for some term } t \text{ with } \mathrm{FV}(t) \subset \mathrm{FV}(A).$$

Proof. By a straightforward induction on the length of derivations. □.

As noted above, for ∃-free formulas there are canonical realizers, and truth and realizability coincide for ∃-free formulas. Therefore the ∃-free formulas of $\mathbf{HA}^\omega$ play the same role w.r.t. $\underline{\mathtt{mr}}$-realizability as the ∃-free formulas of $\mathbf{HA}^*$ w.r.t. $\underline{\mathtt{rn}}$-realizability.
For an axiomatization we need the following

3.6. LEMMA. *For each instance F of one of the following schemata*

$\mathrm{IP}_{\mathrm{ef}}$	$(A \to \exists x^\sigma B) \to \exists y^\sigma (A \to B)$ $\;(y \notin \mathrm{FV}(A),\ A\ \exists\text{-free})$,
AC	$\forall x^\sigma \exists y^\tau A(x,y) \to \exists z^{\sigma\tau} \forall x^\sigma A(x, zx)$,

there is a term t such that $\vdash t\,\underline{\mathtt{mr}}\,F$, with $\mathrm{FV}(t) \subset \mathrm{FV}(A)$.

3.7. THEOREM. *(Axiomatization of modified realizability)*

$$\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}} \vdash A \leftrightarrow \exists x (x\,\underline{\mathtt{mr}}\,A)$$

and for $\mathbf{H} \in \{\mathbf{HA}^\omega, \mathbf{I\text{-}HA}^\omega, \mathbf{E\text{-}HA}^\omega\}$

$$\mathbf{H} + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}} \vdash A \Leftrightarrow \mathbf{H} \vdash t\,\underline{\mathtt{mr}}\,A$$

for some t with $\mathrm{FV}(t) \subset \mathrm{FV}(A) \setminus \{x\}$.

REMARK. In a theory $\mathbf{T}$ with decidable prime formulas $\mathrm{IP}_{\mathrm{ef}}$ is implied by the schema IP defined in 1.13. To see this, note that in $\mathbf{T}$ ∃-free formulas are logically equivalent to negated formulas, since $\neg\neg B \leftrightarrow B$ (by induction on the construction of B). On the other hand, IP is $\underline{\mathtt{mr}}$-realizable in $\mathbf{HA}^\omega$, so the preceding theorem also holds with IP replacing $\mathrm{IP}_{\mathrm{ef}}$, for $\mathbf{H}$ equal to $\mathbf{I\text{-}HA}^\omega$ or $\mathbf{E\text{-}HA}^\omega$.

3.8. THEOREM. *(Applications of modified realizability) Let* $\mathbf{H} \in \{\mathbf{HA}^\omega, \mathbf{I\text{-}HA}^\omega, \mathbf{E\text{-}HA}^\omega\}$, *and let* $\mathbf{H}'$ *be* $\mathbf{H} \pm \mathrm{IP}_{\mathrm{ef}} \pm \mathrm{AC}$. *Then*

(i) $\mathbf{H}'$ *is consistent.*

(ii) $\mathbf{H}' \vdash A \vee B \Rightarrow \mathbf{H}' \vdash A$ *or* $\mathbf{H}' \vdash B$ *(for $A \vee B$ closed) (Disjunction Property DP).*

(iii) $\mathbf{H}' \vdash \exists x^\sigma A \Rightarrow \mathbf{H}' \vdash A[x/t^\sigma]$ *for a suitable term t,* $\mathrm{FV}(t) \subset \mathrm{FV}(A) \setminus \{x\}$ *(Explicit Definability ED).*

(iv) $\mathbf{H}' \vdash \forall x^\sigma \exists y^\tau A(x,y) \Rightarrow \mathbf{H}' \vdash \exists z^{\sigma\tau} \forall x^\sigma A(x, zx)$ *(Rule of choice* ACR*).*

(v) $\mathbf{H}' \vdash (A \to \exists x^\sigma B) \Rightarrow \mathbf{H}' \vdash \exists x^\sigma (A \to B)$ *where A is ∃-free* ($\mathrm{IPR}_{\mathrm{ef}}$*-rule).*

3.9. *Concrete forms of modified realizability*

The proof-theoretic applications of $\underline{\mathtt{mr}}$-realizability obtained by specifying a model for $\mathbf{HA}^\omega$ have in fact two "levels of freedom": (a) the choice of a model $\mathcal{M}$, definable in a language $\mathcal{L}$ say, and (b) the theory formulated in $\mathcal{L}$ which is available for proving facts about $\mathcal{M}$, i.e. the metatheory for $\mathcal{M}$.

By "$\mathcal{M}$ definable in $\mathcal{L}$" we do not mean that $\mathcal{M}$ is globally definable in $\mathcal{L}$, but only that locally, for each A of $\mathbf{HA}^\omega$, we can express $[\![A]\!]_\mathcal{M}$ by a formula of $\mathcal{L}$. Thus choosing HRO for $\mathcal{M}$ is the first level of freedom, and choosing some theory Γ in the language of $\mathbf{HA}^*$ for proving facts about HRO is the second level of freedom.

An interesting example of this occurs in connection with two models of $\mathbf{HA}^\omega$ which are similar to HRO and HEO respectively, but based on partial continuous function application $|$ instead of partial recursive application $\bullet$.

The *Intensional Continuous Functionals* ICF are an analogue of HRO; we give the intuitively simplest definition (which does not mean the technically slickest) of the types:

$$\begin{aligned}
\mathrm{ICF}_0 &:= \mathbb{N},\\
\mathrm{ICF}_{00} &:= \mathbb{N} \to \mathbb{N},\\
\mathrm{ICF}_{\sigma 0} &:= \{\alpha \,:\, \forall\beta{\in}\mathrm{ICF}_\sigma(\alpha(\beta)\downarrow)\}\ (\sigma \neq 0),\\
\mathrm{ICF}_{0\sigma} &:= \{\alpha \,:\, \forall x(\lambda n.\alpha(\langle x\rangle * n) \in \mathrm{ICF}_\sigma)\}\ (\sigma \neq 0),\\
\mathrm{ICF}_{\sigma\tau} &:= \{\alpha \,:\, \forall\beta{\in}\mathrm{ICF}_\sigma(\alpha|\beta \in \mathrm{ICF}_\tau)\}\ (\sigma, \tau \neq 0).
\end{aligned}$$

Application is then defined in the obvious way: $\mathrm{App}_{\sigma,0}(\alpha, \beta) := \alpha(\beta)$, $\mathrm{App}_{0,\sigma}(\alpha, n) := \lambda m.\alpha(\langle n\rangle * m)$, $\mathrm{App}_{\sigma,\tau}(\alpha, \beta) := \alpha|\beta$, etc. Equality at type σ is interpreted by equality of numbers (for $\sigma = 0$) or functions (for $\sigma \neq 0$).

The *Extensional Continuous Functionals* ECF are related to ICF in the same way as HEO is related to HRO: one defines a hereditary equivalence relation based on $|$ instead of $\bullet$. ECF coincides with Kleene'sKleene, S.C. countable functionals or Kreisel'sKreisel, G. continuous functionals.

Both ICF and ECF are locally definable in the language of $\mathbf{EL}^*$, and for soundness of ICF-$\underline{\mathtt{mr}}$ and ECF-$\underline{\mathtt{mr}}$ relative to $\mathbf{EL}^*$ nothing more is needed. But additional axioms added to $\mathbf{EL}^*$ may result in different properties of the models, and hence of $\mathcal{M}$-$\underline{\mathtt{mr}}$-realizability. Two mutually incompatible additional axioms we can add to $\mathbf{EL}^*$ are $\mathrm{FAN_D}$ and a version of *Church's Thesis*Church, A.

CT $\qquad \forall\alpha\exists x\forall y(\alpha x = x{\bullet}y).$

CT states that the function variables in $\mathbf{EL}^*$ range over the total recursive functions; the incompatibility of CT with $\mathrm{FAN_D}$ follows from Kleene'sKleene, S.C. well-known example of a primitive recursive tree well-founded w.r.t. all total recursive functions but not w.r.t. all functions, since the depth of the tree is unbounded (cf. (**?**, 4.7.6)).

Assuming $\mathrm{FAN_D}$, we can show that ICF and ECF contain a *Fan Functional* $\phi_{\rm uc}$ satisfying the axiom for a *Modulus of Uniform Continuity*

$$\text{MUC} \qquad \forall z^2 \forall\gamma\forall\alpha{\le}\gamma\forall\beta{\le}\gamma(\bar\alpha(\phi_{\rm uc}z\gamma) = \bar\beta(\phi_{\rm uc}z\gamma) \to z\alpha = z\beta).$$

If we add MUC to $\mathbf{HA}^\omega$, we can $\underline{\underline{\texttt{mr}}}$-interpret $\mathrm{FAN_D}$. If, on the other hand, we use $\mathbf{EL}^* + \mathrm{CT}$ as our metatheory for ICF-$\underline{\underline{\texttt{mr}}}$, we can realize a statement positively contradicting MUC. See (**?**, 2.6.4, 2.6.6, 3.4.16, 3.4.19).

As an example of an application of a concrete version of $\underline{\underline{\texttt{mr}}}$-realizability we can show e.g. the consistency of $\mathbf{HA}^\omega$ + $\mathrm{IP_{ef}}$ + AC + WC-N + $\mathrm{FAN_D}$ + $\mathrm{EXT_{1,0}}$, where WC-N is the schema $\forall\alpha\exists n\, A(\alpha, n) \to \forall\alpha\exists n, m\forall\beta(\bar\alpha m = \bar\beta m \to A(\beta, n))$, and $\mathrm{EXT_{1,0}}$ is $\forall\alpha\beta z^2(\alpha = \beta \to z^2\alpha = z^2\beta)$. (Use ICF-$\underline{\underline{\texttt{mr}}}$-realizability with $\mathbf{EL}^*$ + $\mathrm{FAN_D}$ as metatheory.)

Notation. Henceforth we write $\underline{\underline{\texttt{mrn}}}$, $\underline{\underline{\texttt{mrf}}}$ for HRO-$\underline{\underline{\texttt{mr}}}$ and ICF-$\underline{\underline{\texttt{mr}}}$-realizability respectively. □.

3.10. *Notes*

Modified realizability was first formulated by (**?**); a concrete version equivalent to our ICF-$\underline{\underline{\texttt{mr}}}$-realizability was used in (**?**).

(**?**) and (**?**) apply $\underline{\underline{\texttt{mrt}}}$-realizability to bounded arithmetic and related systems, improving on earlier results obtained by (**?**) by means of numerical realizability.

(**?**) used modified realizability to obtain consistency of intuitionistic analysis with a restricted form of IP (Vesley's principle). (**?**) used a modified realizability interpretation to obtain consistency of a weak version of Church'sChurch, A. thesis with Kleene'sKleene, S.C. system for intuitionistic analysis (i.e **EL** with bar induction and GC for the case $A \equiv 0 = 0$), together with Vesley's Principle. The weak version of Church'sChurch, A. thesis may be stated as: "each numerical function is not not recursive". In (**?**, 3.4.15) it is observed that the modified realizability of (**?**) is essentially abstract modified realizability interpreted in the type structure consisting of the *recursive* elements of ICF, and that the consistency proof covers in fact full $\mathrm{IP_{ef}}$ ((**?**, 3.4.18)).

Some further examples of papers using or discussing modified realizability are Dragalin(**?**), Diller(**?**), Grayson(**?**, **?**), van Oosten (**?**). Scedrov and Vesley (**?**). See also 9.8 on (**?**, **?**).

4 Derivation of the Fan Rule

This section is devoted to an "indirect application" of modified realizability: it is shown how closure under the rule of choice ACR, obtained from $\underline{\underline{\texttt{mrt}}}$-realizability, may be combined with the (intrinsically interesting) notion of "majorizable functional" to obtain closure under the Fan Rule.

We can define the so-called *majorizable* functionals relative to any finite-type structure. They are introduced via a relation of majorization, defined as follows.

4.1. Definition. $t^* \operatorname{maj}_\sigma t$, for $t^*, t \in \sigma$, is defined by induction on σ:

$$\begin{aligned}
&t^* \operatorname{maj}_0 t &&:= t^* \geq t,\\
&t^* \operatorname{maj}_{\sigma\times\tau} t &&:= \mathbf{p}_0 t^* \operatorname{maj}_\sigma \mathbf{p}_0 t \wedge \mathbf{p}_1 t^* \operatorname{maj}_\tau \mathbf{p}_1 t,\\
&t^* \operatorname{maj}_{\tau\sigma} t &&:= \forall y^* y (y^* \operatorname{maj}_\tau y \to t^* y^* \operatorname{maj}_\sigma ty, t^* y).
\end{aligned}$$

Furthermore we put

$$t \in \mathrm{Maj} := \exists t^* \operatorname{maj}_\sigma t \ \ (\text{“}t \text{ is majorizable”}).$$

Lemma. $t^* \operatorname{maj} t \Rightarrow t^* \operatorname{maj} t^*$.
Proof. Induction on the type of t.

4.2. Definition. For each $t \in 0\sigma$ we define $t^+ \in 0\sigma$ by induction on the structure of σ.

$$\begin{aligned}
&t^+ 0 = t0,\ t^+(Sz) = \max\{t^+ z, t(Sz)\} \ \text{ for } \sigma = 0,\\
&t^+ := \lambda n.[\lambda y((\lambda n.tny)^+ n)] \ \text{ for } \sigma \equiv \sigma_1\sigma_2,\\
&t^+ = \lambda n.\mathbf{p}((\lambda n.\mathbf{p}_0(tn))^+ n)((\lambda n.\mathbf{p}_1(tn))^+ n) \ \text{ for } \sigma \equiv \sigma_1 \times \sigma_2.
\end{aligned}$$

Lemma. *If* $\forall n^0 (Fn \operatorname{maj} Gn)$, *then* $F^+ \operatorname{maj} G^+, G$.
Proof. We use induction on σ. Let $Fn, Gn \in \sigma$.
Case (i) $\sigma \equiv 0$. Almost immediate.
Case (ii) $\sigma \equiv \sigma_1\sigma_2$. The assumption yields

$$s^* \operatorname{maj} s \Rightarrow Fns^* \operatorname{maj} Fns, Gns$$

for all $n \in \mathbb{N}$. By the induction hypothesis we have

(1) $$(\lambda n.Fns^*)^+ \operatorname{maj} (\lambda n.Fns)^+, (\lambda n.Fns), (\lambda n.Gns)^+, (\lambda n.Gns),$$

Now by definition of F^+, G^+ and beta-conversion:

$$\begin{aligned}
&(\lambda n.Fns^*)^+ k = F^+ ks^*\\
&(\lambda n.Fns)^+ k = F^+ ks\\
&(\lambda n.Gns)^+ k = G^+ ks
\end{aligned}$$

If $n \geq m$, we obtain from (1)

$$F^+ ns^* \operatorname{maj} F^+ ns, F^+ ms, Fms, \quad F^+ ns^* \operatorname{maj} G^+ ms, Gms.$$

and from this $F^+ n \operatorname{maj} F^+ m, Fm, G^+ m, Gm$. Since $n \geq m$, it follows that $F^+ \operatorname{maj} G^+, G$.

Case (iii) $\sigma \equiv \sigma_1 \times \sigma_2$. We are given $\forall n(Fn \,\mathrm{maj}\, Gn)$, so

$$\forall n(\mathbf{p}_i(Fn) \,\mathrm{maj}\, \mathbf{p}_i(Gn)) \ (i \in \{0,1\}).$$

So we have

$$\forall n((\lambda n.\mathbf{p}_i(Fn))n \,\mathrm{maj}\, (\lambda n.\mathbf{p}_i(Gn))n)$$

and hence by the induction hypothesis

$$(\lambda n.\mathbf{p}_i(Fn))^+ \,\mathrm{maj}\, (\lambda n.\mathbf{p}_i(Gn))^+, \lambda n.\mathbf{p}_i(Gn).$$

From this we obtain for $n \geq m$, $i \in \{0,1\}$

$$(\lambda n.\mathbf{p}_i(Fn))^+ n \,\mathrm{maj}\, (\lambda n.\mathbf{p}_i(Fn))^+ m, (\lambda n.\mathbf{p}_i(Gn))^+ m, (\lambda n.\mathbf{p}_i(Gn))m,$$

$$\mathbf{p}_i(F^+ n) \,\mathrm{maj}\, \mathbf{p}_i(F^+ m),\ \mathbf{p}_i(G^+ m), \mathbf{p}_i(Gm)$$

and therefore hence $F^+ \,\mathrm{maj}\, G^+, G$.

4.3. Proposition. *Let all free variables in $t \in \tau$ be of type 0 or 1; then there is a term $t^* \in \tau$ with $\mathrm{FV}(t^*) \subset \mathrm{FV}(t)$, such that $\mathbf{HA}^\omega \vdash t^* \,\mathrm{maj}\, t^*, t$.*

Proof. For each constant or variable of type 0 or 1 of $\mathbf{HA}^\omega$ (c^τ say) we show that there is a $c^* \in \tau$ with $c^* \,\mathrm{maj}_\tau\, c$.

(a) $0 \,\mathrm{maj}\, 0$, $S \,\mathrm{maj}\, S$ are immediate;
(b) $x^0 \,\mathrm{maj}\, x^0$; for y^1 define y^* by recursion as y^+;
(c) $\mathbf{k} \,\mathrm{maj}\, \mathbf{k}$, $\mathbf{s} \,\mathrm{maj}\, \mathbf{s}$, $\mathbf{p} \,\mathrm{maj}\, \mathbf{p}$, $\mathbf{p}_0 \,\mathrm{maj}\, \mathbf{p}_0$, $\mathbf{p}_1 \,\mathrm{maj}\, \mathbf{p}_1$;
(d) If $\mathbf{r}$ is the recursor with $\mathbf{r}0ts = t$ etc., take $\mathbf{r}^* := \mathbf{r}^+$.

4.4. Theorem. *(Fan Rule) Let A be a formula of $\mathbf{HA}^\omega$ containing only variables of types 0 or 1 free, then $\mathbf{HA}^\omega \vdash \forall \alpha{\leq}\beta \exists n\, A(\alpha, n) \Rightarrow \mathbf{HA}^\omega \vdash \exists m \forall \alpha{\leq}\beta \exists n{\leq}m\, A(\alpha, n)$, where $\alpha \leq \beta := \forall m(\alpha n \leq \beta n)$.*

Proof. Let $\mathbf{HA}^\omega \vdash \forall \alpha{\leq}\beta\, A(\alpha, F\alpha)$ for a suitable term $F \in (1)0$. F is majorizable, so there is an F^* such that $F^* \,\mathrm{maj}\, F^*, F$ which means in particular that $\forall \alpha\beta(\beta \geq \alpha \to F^*\beta^+ \geq F\alpha)$ and hence $\mathbf{HA}^\omega \vdash \forall \alpha{\leq}\beta \exists n{\leq}F^*\beta^+\, A(\alpha, n)$. □

Remarks. Switching from a recursor of type $\sigma(\sigma 0 \sigma)0\sigma$ to a recursor of type $0(\sigma 0 \sigma)\sigma\sigma$ is purely a matter of technical convenience; these recursors are interdefinable.

In (**?**) the following generalization is established for $\mathbf{E\text{-}HA}^\omega$:

$$\vdash \forall \alpha \forall x{\leq_\rho} s\alpha \exists y^\tau A(\alpha, x, y) \Rightarrow \vdash \forall \alpha \forall x{\leq_\rho} s\alpha \exists y{\leq_\tau} \alpha A(\alpha, x, y),$$

where $\tau \in \{0,1,2\}$, $s \in 1\rho$, s closed, and where $\leq_\sigma$ is defined by induction on the type structure by $x^0 \leq_0 y_0 := x \leq y$, $x^{\sigma\tau} \leq_{\sigma\tau} y^{\sigma\tau} := \forall z^\sigma(xy \leq_\tau yz)$, $x^{\sigma\times\tau} \leq_{\sigma\times\tau} y^{\sigma\times\tau} := \mathbf{p}_0 x \leq_\sigma \mathbf{p}_0 y \wedge \mathbf{p}_1 x \leq_\tau \mathbf{p}_1 y$.

As observed above, the proof of closure under the Fan Rule given above depends on realizability only to the extent that we have used modified realizability to obtain closure under the fan rule. For other systems other interpretations, such as the Dialectica interpretation, yield closure under the rule of choice; cf. (**?**).

4.5. *Notes*

The notions of *majorization* and *majorizable functional* were introduced by (**?**). The present version is a modification due to Bezem(**?**, **?**), called strong majorization by him; we have added a clause for product types.

(**?**) introduced a version of Bezem'sBezem, M. definition with a special clause for types of the form $\sigma 0$; however, in the presence of product types we found it more convenient to stick to Bezem'sBezem, M. definition.

The proof of the Fan Rule presented here is due to (**?**). For other proofs, see e.g. (**?**), (**?**), (**?**, 9.7.23).

5 Lifschitz realizability

Lifschitz, V. This type of realizability was invented by (**?**) to show that *Church's Thesis with Uniqueness*Church, A.

$\mathrm{CT}_0!$ $\quad \forall x \exists! y\, A(x,y) \to \exists z \forall x (z{\bullet}x{\downarrow} \wedge A(x, z{\bullet}x))$

does not imply CT_0 in $\mathbf{HA}^*$. The idea to achieve this, is to use as realizer for an existential formula not a single instantiation for the quantifier, but a finite inhabited set of possible instantiations, such that in general there is no recursive procedure for selecting elements of such inhabited sets, although for singletons there is such a procedure. The sets we use are given by

$$V_x := \{y : y \le \mathbf{p}_1 x \wedge \forall n \neg T(\mathbf{p}_0 x, y, n)\}.$$

If we know that V_x is a singleton, say $\{y : 0 = 0\}$, we can find y recursively in x as follows: we start computing $\mathbf{p}_0 x{\bullet}z$ for all values of $z \le \mathbf{p}_1 x$; as soon as we have found terminating computations for $\mathbf{p}_1 x$ arguments, we know that the remaining argument $\le \mathbf{p}_1 x$ is the required y.

5.1. Definition. The clauses for $\underline{\mathtt{rln}}$-realizability are identical to the clauses for $\underline{\mathtt{rn}}$-realizability, except for the existential quantifier:

$$x\, \underline{\mathtt{rln}}\, \exists y\, A \;:=\; \mathrm{Inh}(V_x) \wedge \forall y \in V_x (\mathbf{p}_1 y\, \underline{\mathtt{rln}}\, A[y/\mathbf{p}_0 y])$$

where "Inh(W)" means that $\exists z (z \in W)$. □

In this form the notion appears as a modification of numerical realizability. There is also a Lifschitz'sLifschitz, V. analogue of function realizability. In that case the sets of realizers for the existential quantifiers take the form

$$V_\alpha := \{\gamma : \gamma \le \mathbf{p}_1 \alpha \wedge \forall n (\mathbf{p}_0 \alpha(\bar{\gamma} n) = 0)\}.$$

The V_α are not finite, but compact. There is no general method for finding an element in inhabited V_α which is continuous in α, but there is a method for V_α's which are singletons. There is no interesting "abstract" version of LifschitzLifschitz, V. realizability.

5.2. DEFINITION. $\underline{\texttt{rlf}}$-realizability is defined as $\underline{\texttt{rf}}$-realizability, except for the clauses for the existential quantifiers, which become:

$$\alpha\,\underline{\texttt{rlf}}\,\exists\beta A := \mathrm{Inh}(V_\alpha) \wedge \forall\gamma \in V_\alpha(\mathbf{p}_1\gamma\,\underline{\texttt{rlf}}\,A[\beta/\mathbf{p}_0\gamma]),$$
$$\alpha\,\underline{\texttt{rlf}}\,\exists x A := \mathrm{Inh}(V_\alpha) \wedge \forall\gamma \in V_\alpha(\mathbf{p}_1\gamma\,\underline{\texttt{rlf}}\,A[x/(\mathbf{p}_0\gamma)0]). \quad \Box$$

5.3. *Summary of results for* $\underline{\texttt{rln}}$*-realizability*

DEFINITION. In $\mathbf{HA}^*$ the *bounded* Σ^0_2*-formulas* ($\mathrm{B}\Sigma^0_2$-formulas) are formulas of the form $\exists x{<}t\neg(s = s')$; the $\mathrm{B}\Sigma^0_2$*-negative formulas* are the formulas constructed from prime formulas $s = s'$ and $\mathrm{B}\Sigma^0_2$-formulas by means of $\forall, \wedge, \rightarrow$. $\Box$

Corresponding classes in $\mathbf{HA}$ are defined as follows. A formula of the form $\exists x{\leq}y\forall z\, A$ with A primitive recursive is called a *bounded* Σ^0_2*-formula* ($\mathrm{B}\Sigma^0_2$-formula); the $\mathrm{B}\Sigma^0_2$*-negative formulas* are the formulas constructed from Σ^0_1-formulas and $\mathrm{B}\Sigma^0_2$-formulas by means of $\forall$, $\wedge$, $\rightarrow$.
N.B. Although the class of $\mathrm{B}\Sigma^0_2$-formulas in $\mathbf{HA}^*$ is somewhat wider than the corresponding class in $\mathbf{HA}$, the $\mathrm{B}\Sigma^0_2$-negative formulas for $\mathbf{HA}^*$ and $\mathbf{HA}$ are the same modulo logical equivalence. To see this, observe that (a) a Π^0_1-formula in $\mathbf{HA}$ can be written as $\neg s = s$ in $\mathbf{HA}^*$, and (b) $\exists x < t\neg(s = s')$ in $\mathbf{HA}^*$ is equivalent to a formula of the form $\exists y(t = y) \wedge \forall z(t = z \rightarrow \exists x{<}z.A(x))$, with A primitive recursive, which is $\mathrm{B}\Sigma^0_2$-negative in $\mathbf{HA}$ modulo logical equivalence.

In the case of numerical LifschitzLifschitz, V. realizability, we cannot take as our basis theory $\mathbf{HA}^*$, but need instead an extension $\mathbf{HA'}$, which is $\mathbf{HA}^* + \mathrm{M} + \mathrm{CB}\Sigma^0_2$; here $\mathrm{CB}\Sigma^0_2$, the *Cancellation of double negations in Bounded* Σ^0_2*-formulas* is:

$\mathrm{CB}\Sigma^0_2$ $\qquad \neg\neg A \rightarrow A$ (for A in $\mathrm{B}\Sigma^0_2$).

Van OostenOosten, J. van showed that in fact $\mathrm{CB}\Sigma^0_2$ is equivalent to the following principle

$$\forall nm(Pn \vee Qm) \rightarrow (\forall nPn \vee \forall mQm) \;\; (P, Q \text{ primitive recursive}),$$

where $n \notin \mathrm{FV}(Q)$, $m \notin \mathrm{FV}(P)$. Soundness now holds w.r.t. $\mathbf{HA'}$, i.e. for all sentences A

$$\mathbf{HA'} \vdash A \;\Rightarrow\; \mathbf{HA'} \vdash \bar{n}\,\underline{\texttt{rln}}\,A$$

for a suitable numeral $\bar{n}$. The following properties of the V_n are crucial in the proof of the soundness theorem:

(i) for some total recursive f_0, $\forall xy(y \in V_{f_0(x)} \leftrightarrow y = x)$, i.e. indices of singleton V_t's may be found recursively in their (unique) elements.

(ii) There is a partial recursive f_1 such that for any operation with code x, total on V_y, the image of V_y under x is $V_{f_1(x,y)}$.

(iii) There is a total recursive f_2 such that $V_{f_2(x)} = \bigcup\{V_z \,:\, z \in V_x\}$.

(iv) There is a partial recursive f_3 such that $\mathbf{HA}' \vdash \forall x(\mathrm{Inh}(V_x) \wedge \forall y{\in}V_x(y\,\underline{\mathtt{rln}}\,A) \rightarrow f_3(x)\,\underline{\mathtt{rln}}\,A)$.

With respect to the class of self-realizing formulas, we note an interesting deviation from the notions of realizability considered hitherto: these are not just the $\exists$-free formulas, but the wider class of $\mathrm{B}\Sigma^0_2$-negative formulas. Now we can axiomatize $\underline{\mathtt{rln}}$-realizability relative to $\mathbf{HA}'$ by means of the following scheme for $\mathrm{B}\Sigma^0_2$-negative A:

$\mathrm{ECT_L}$ $\quad \forall x(Ax \rightarrow \exists y Bxy) \rightarrow \exists z \forall x(Ax \rightarrow z{\bullet}x{\downarrow} \wedge \mathrm{Inh}(V_{z\bullet x}) \wedge \forall u{\in}V_{z\bullet x}\, Bxu)$.

An interesting special case of $\mathrm{ECT_L}$ is $\mathrm{ECT_L}!$ which can be formulated as

$$\forall x(Ax{\rightarrow}\exists! y\, Bxy) \rightarrow \exists z \forall x(Ax \rightarrow z{\bullet}x{\downarrow}{\wedge}B(x, z{\bullet}x)) \quad (A\ \mathrm{B}\Sigma^0_2\text{-negative}),$$

with the help of the following

Lemma. *There is a partial recursive f_5 such that*

$$\mathbf{HA}' \vdash \forall z(\exists x \forall y(x = y \leftrightarrow y \in V_z) \rightarrow f_5(z) \in V_z).$$

5.4. Proposition. *(Applications)*

(i) $\mathbf{HA}' + \mathrm{ECT_L}$ *is consistent;*

(ii) $\mathbf{HA}^* + \mathrm{ECT_L}! \nvdash \mathrm{CT}_0$*;*

(iii) $\mathbf{HA}'$ *is closed under the rule* $\mathrm{ECR_L}$ *and a fortiori under the rule* $\mathrm{ECR_L}!$ *(as* $\mathrm{ECT_L}$ *and* $\mathrm{ECT_L}!$ *but with main* $\rightarrow$ *replaced by* $\Rightarrow$ *etc). Since* $\mathbf{HA}'$ *also satisfies the rules* DP *and* EDN*, we can formulate the rule* $\mathrm{ECR_L}!$ *even more strongly as: for A in* $\mathrm{B}\Sigma^0_2$*,*

$$\vdash \forall x(Ax \rightarrow \exists! y Bxy) \Rightarrow \vdash \forall x(Ax \rightarrow \bar{n}{\bullet}x{\downarrow} \wedge B(x, \bar{n}{\bullet}x)$$

for a suitable numeral $\bar{n}$.

5.5. *Summary of results for* $\underline{\mathtt{rlf}}$*- and* $\underline{\mathtt{rlft}}$*-realizability*

The basis theory is now an extension of $\mathbf{EL}^*$, namely $\mathbf{EL}' \equiv \mathbf{EL}^* + \mathrm{M_{QF}} + \mathrm{KL_{QF}}$, where $\mathrm{M_{QF}}$ is Markov's principle for quantifier-free formulas, and *König'sKönig, J. Lemma for quantifier-free formulas* is the schema for *quantifier-free A.*

$$\mathrm{KL_{QF}} \quad \begin{array}{l} \forall x \exists n(\mathrm{lth}(n) = x \wedge n{\leq}\alpha \wedge An) \\ \qquad \wedge \forall nm(A(n * m) \rightarrow An) \rightarrow \exists \beta{\leq}\alpha \forall n\, A(\bar{\beta}n) \end{array}$$

$(n{\leq}\alpha := \forall y{<}\mathrm{lth}(n)((n)_y \leq \alpha y)$; $\exists \alpha{\leq}\phi(\ldots) := \exists \alpha(\alpha \leq \phi \wedge \ldots))$. The analogue in $\mathbf{EL}'$ of the $\mathrm{B}\Sigma^0_2$-negative formulas are the $\mathrm{B}\Sigma^1_2$-negative formulas (a $\mathrm{B}\Sigma^1_2$-formula is a formula of the form $\exists \alpha{\leq}\phi \neg s = t$ (a *Bounded* Σ^1_2 *formula*);

the class of $B\Sigma^1_2$-*negative formulas* is obtained from formulas $B\Sigma^1_2$-formulas and prime formulas by means of $\rightarrow$, $\wedge$, $\forall$). $\Box$

As a typical result one obtains that GC! (i.e.the special case of GC with uniqueness for the existential quantifier) does not imply GC, not even the special case of WC-N.

REMARK. $\mathrm{KL_{QF}}$ for the language of $\mathbf{EL}^*$ follows from $\mathrm{KL_{QF}}$ for **EL** by observing that KL with $A \in \Sigma^0_1$ is derivable from $\mathrm{KL_{QF}}$ in **EL**.

5.6. *Notes*

(**?**) defined Lifschitz'Lifschitz, V. realizability for certain set theories and uses it to obtain independence of CT_0 from $\mathrm{CT}_0!$ for these theories. Other relevant papers are van Oosten(**?**, **?**, **?**). As to (**?**), see 8.28.

It is possible to combine LifschitzLifschitz, V. realizability with modified realizability for HRO (**?**).

It is not known whether for some or all results perhaps weaker theories than $\mathbf{HA}'$, $\mathbf{EL}'$ will suffice.

6 Extensional realizability

It is also possible to combine the idea of realizability with extensionality, by defining not just a notion of the form "x realizes A", but a relation between realizing objects: "x and y equally realize A". The definition below has been written out for $\mathbf{HA}^*$ and partial recursive application, but also makes sense in the abstract setting of **APP**, if we read everywhere $\underline{\mathtt{re}}$ for $\underline{\mathtt{rne}}$.

6.1. DEFINITION. We define "$x = x' \,\underline{\mathtt{rne}}\, A$" ($x, x' \notin \mathrm{FV}(A)$, $x \not\equiv y$), by induction on the complexity of A:

$$\begin{aligned}
x = x' \,\underline{\mathtt{rne}}\, P &:= (x = x' \wedge P \wedge x{\downarrow} \wedge x'{\downarrow}) \ \ (P \text{ prime}),\\
x = x' \,\underline{\mathtt{rne}}\, (A \wedge B) &:= (\mathbf{p}_0 x = \mathbf{p}_0 x' \,\underline{\mathtt{rne}}\, A) \wedge (\mathbf{p}_1 x = \mathbf{p}_1 x' \,\underline{\mathtt{rne}}\, B),\\
x = x' \,\underline{\mathtt{rne}}\, (A \rightarrow B) &:= x{\downarrow} \wedge x'{\downarrow} \wedge \forall y y'(y = y' \,\underline{\mathtt{rne}}\, A \rightarrow\\
&\quad x \bullet y = x \bullet y' \,\underline{\mathtt{rne}}\, B \wedge x' \bullet y = x' \bullet y' \,\underline{\mathtt{rne}}\, B \wedge x \bullet y = x' \bullet y \,\underline{\mathtt{rne}}\, B),\\
x = x' \,\underline{\mathtt{rne}}\, \forall y\, A &:= \forall y (x \bullet y = x' \bullet y \,\underline{\mathtt{rne}}\, A),\\
x = x' \,\underline{\mathtt{rne}}\, \exists y\, A &:= (\mathbf{p}_0 x = \mathbf{p}_0 x') \wedge (\mathbf{p}_1 x = \mathbf{p}_1 x' \,\underline{\mathtt{rne}}\, A[y/\mathbf{p}_0 x]),
\end{aligned}$$

and we put

$$x \,\underline{\mathtt{rne}}\, A \ := \ x = x \,\underline{\mathtt{rne}}\, A.$$

As always, $\underline{\mathtt{rnet}}$-realizability is obtained by adding "$\wedge\ (A \rightarrow B)$" in the implication clause. $\Box$

REMARKS. Note that $x\,\underline{\mathtt{rn}}\,A$ does not in general imply $x = x\,\underline{\mathtt{rne}}\,A$; for if $x = x\,\underline{\mathtt{rne}}\,[\forall y(t = 0) \to \forall z(s = 0)]$, then x must yield the same value when applied to extensionally equal realizers z, z' for $\forall y(t = 0)$; on the other hand, for an x such that $x\,\underline{\mathtt{rn}}\,[\forall y(t = 0) \to \forall z(s = 0)]$ no such restriction applies.

The definition may also be formulated as a simultaneous inductive definition of "x extensionally realizes A" and "x and y are equivalent realizers for A", but this is more cumbersome.

It is straightforward to prove soundness.

The $\exists$-free formulas play the same role as in $\underline{\mathtt{rn}}$-realizability. On the other hand, no simple axiomatization of the provably $\underline{\mathtt{rne}}$-realizable formulas is known.

For proofs of the following facts we refer to (**?**).

6.2. The difference between ordinary realizability and extensional realizability is demonstrated by the fact that the following instance of ECT_0 is not $\underline{\mathtt{rne}}$-realizable:

$$\forall z[\forall x \exists y(\neg\neg\exists u Tzxu \to Tzxy) \to \exists v \forall x(v{\bullet}x \wedge (\neg\neg\exists u Tzxu \to T(z, x, v{\bullet}x))]$$

On the other hand it is not hard to verify that the following *"Weak Extended Church's Thesis"* Church, A. is provably $\underline{\mathtt{rne}}$-realizable:

6.3. PROPOSITION. *In* **HA** *we can* $\underline{\mathtt{rne}}$*-realize:*

WECT_0 $\quad \forall x(A \to \exists y\, Bxy) \to \neg\neg\exists z \forall x(A \to z{\bullet}x{\downarrow} \wedge B(x, z{\bullet}x))$

for $\exists$*-free* A.

A nice application of $\underline{\mathtt{rnet}}$-realizability is the following refinement of ECR.

6.4. PROPOSITION. *Assume for* $\exists$*-free* B *that in* $\mathbf{HA}^*$

$$\vdash \forall z(\forall x \exists y\, Bzxy \to \exists u\, Czu)$$

then for some $\bar{n}$

$$\vdash \forall z(\bar{n}{\bullet}z{\downarrow} \wedge \forall v, v'(\forall x(v{\bullet}x = v'{\bullet}x \wedge B(z, x, v{\bullet}x)) \to \bar{n}{\bullet}z{\bullet}v = \bar{n}{\bullet}z{\bullet}v' \wedge C(z, \bar{n}{\bullet}z{\bullet}v)).$$

6.5. *Notes*

Extensional realizability appears for the first time explicitly in some unpublished notes by (**?**), and implicitly in (**?**).

(**?**) and Beeson(**?**, **?**) use an abstract version of extensional realizability in combination with forcing, to prove that $\mathbf{ML}_0$ (the arithmetical fragment of the extensional version of Martin-Löf'sMartin-Löf, P. type theory) is conservative over **HA**. $\mathbf{ML}_0$ includes $\mathbf{E}\text{-}\mathbf{HA}^\omega$+AC as a subtheory. See also (**?**). The proofs by RenardelRenardel de Lavalette, G. R. and BeesonBeeson, M. J. extend earlier work of (**?**)[3].

There is a close similarity between "$x = x' \,\underline{\texttt{rne}}\, A$" and "$x = x' \in A$" in the type-theories of Martin-LöfMartin-Löf, P. (**?**, **?**), so it is not surprising that an interpretation akin to extensional realizability can be used to model (parts of) Martin-Löf'sMartin-Löf, P. extensional type theories, cf. (**?**). See also 9.3.

$\underline{\texttt{rne}}$-realizability for the language of arithmetic does not lend itself to a straightforward axiomatization in the same manner as ECT_0 might be said to axiomatize $\underline{\texttt{rn}}$-realizability relative to **HA**. But (**?**) showed that axiomatization is possible in a suitably chosen conservative extension of **HA** plus Markov's principle. The same paper discusses also $\underline{\texttt{rne}}$-realizability for higher-order logic in the form of certain toposes.

7 Realizability for intuitionistic second-order arithmetic

7.1. *The system* **HAS**

HAS (*HeytingHeyting, A. Arithmetic of Second order*) is a two-sorted extension of **HA** with quantifiers over $\mathcal{P}(\mathbb{N})$, the powerset of $\mathbb{N}$. So the language of **HA** is extended with set variables X, Y, Z, and corresponding (second-order) quantifiers $\forall X$, $\exists Y$; atomic formulas are now of the form $t = s$ or Xt (also written $t \in X$) for individual terms t, s and set variable X.

Instead of formally introducing set-terms $\lambda x.B$ (B any formula) we can formulate the axiom for second-order $\forall$ as

$$\forall X.A \to A[X/\lambda x.B]$$

where $A[X/\lambda x.B]$ is obtained from A by replacing every occurrence of Xt by $B[x/t]$. Alternatively, we restrict the $\forall^2$-axiom to

$$\forall X.A \to A[X/Y]$$

while adding the axiom schema of *full comprehension*

CA $\qquad \exists X \forall x (Xx \leftrightarrow A) \quad (X \notin \mathrm{FV}(A)).$

Moreover, we require sets to respect equality

$$\forall Xxy(Xx \wedge x = y \to Xy).$$

$\mathbf{HAS}^*$ is related to **HAS** in the same way as $\mathbf{HA}^*$ to **HA**. In particular, for the set variables we have strictness: $Xt \to t\downarrow$.

[3] We do not know whether the treatment in (**?**) is really equivalent to the one in (**?**).

7.2. *Realizability for* $\mathbf{HAS}^*$

It is quite easy to extend $\underline{\mathtt{rn}}$-realizability from $\mathbf{HA}^*$ to $\mathbf{HAS}^*$ by "brute force"; we assign to each set variable X a new set variable X^*, representing the "realizability predicate" and then add the following clauses to $\underline{\mathtt{rn}}$-realizability for $\mathbf{HA}^*$:

$$\begin{aligned} x\,\underline{\mathtt{rn}}\,Xt &:= X^*(x,t) \ \ (x\!\downarrow \text{is automatic by strictness}) \\ x\,\underline{\mathtt{rn}}\,\forall X\,A &:= \forall X^*(x\,\underline{\mathtt{rn}}\,A), \\ x\,\underline{\mathtt{rn}}\,\exists X\,A &:= \exists X^*(x\,\underline{\mathtt{rn}}\,A). \end{aligned}$$

Here $Y(t,t')$ for any set variable Y abbreviates $Y(\mathbf{p}(t,t'))$. (Nothing prevents us from taking $X^* \equiv X$, but in discussions this is sometimes inconvenient and confusing.)

7.3. Remark. In a second-order context, $\bot$, $\exists$ and $\wedge$ are definable in terms of $\to$ and $\forall$, in particular

$$\begin{aligned} \exists Y.A &:= \forall Z^0(\forall Y(A \to Z^0) \to Z^0, \\ A \wedge B &:= \forall Z^0(((A \to (B \to Z)) \to Z), \\ \bot &:= \forall Z^0.Z, \end{aligned}$$

where Z^0 ranges over propositions. (Strictly speaking, we do not have variables over propositions, only over sets, but the addition of proposition variables is conservative, since one may render $(\mathrm{Q}\,Z^0)A(Z^0)$ as $(\mathrm{Q}\,X)A(X0)$ for $\mathrm{Q} \in \{\forall,\exists\}$.) Using this definition of $\exists$, the clause for realizing $\exists X.A$ is in fact redundant, and we obtain an equivalent notion of realizability. A virtually immediate consequence of soundness for $\underline{\mathtt{rn}}$-realizability for $\mathbf{HAS}$ is the consistency of $\mathbf{HAS}$ with Church'sChurch, A. thesis and the so-called *Uniformity principle*

$$\mathrm{UP} \qquad \forall X \exists y\, A(X,y) \to \exists y \forall X\, A(X,y).$$

7.4. Proposition. $\mathbf{HAS}^* + \mathrm{ECT}_0 + \mathrm{UP} + \mathrm{M}$ *is consistent.*
Here ECT_0 is formulated as for $\mathbf{HA}^*$, except that A is restricted to $\exists$-free formulas of $\mathbf{HA}^*$, while B is arbitrary.

7.5. $\underline{\mathtt{rnt}}$*-realizability for* $\mathbf{HAS}^*$

Extension of $\underline{\mathtt{rnt}}$-realizability to $\mathbf{HAS}^*$ is similar to the extension of $\underline{\mathtt{rn}}$-realizability, but we have to be slightly more careful: we want to keep track of realizability *and* truth, so we want to associate with an arbitrary set X an arbitrary Y together with its realizability set Z. It is convenient to encode Y and Z into a single set X^*; we put

$$X^{*\mathrm{t}} := \{n : X^*(2n)\}, \quad X^{*\mathrm{r}} := \{n : X^*(2n+1)\}$$

representing the two components of truth and realizability respectively. The new clauses in the definition of $\underline{\mathtt{rnt}}$-realizability now become

$$
\begin{aligned}
x\,\underline{\mathtt{rnt}}\,Xt &:= X^{*\mathrm{t}}(t) \wedge X^{*\mathrm{r}}(x,t),\\
x\,\underline{\mathtt{rnt}}\,\forall X\,A &:= \forall X^{*}(x\,\underline{\mathtt{rnt}}\,A),\\
x\,\underline{\mathtt{rnt}}\,\exists X\,A &:= \exists X^{*}(x\,\underline{\mathtt{rnt}}\,A),\\
x\,\underline{\mathtt{rnt}}\,(A \to B) &:= \forall y(y\,\underline{\mathtt{rnt}}\,A \to x{\bullet}y\,\underline{\mathtt{rnt}}\,B) \wedge (A \to B)^{*},
\end{aligned}
$$

where C^{*} is obtained from C by replacing all occurrences of Yt by $Y^{*\mathrm{t}}(t)$. It is readily verified that for all A with second-order variables contained in $\{X_1, X_2, \ldots X_n\}$

$$
\begin{aligned}
&\vdash A[X_1, \ldots, X_n/X_1^{*\mathrm{t}}, \ldots, X_n^{*\mathrm{t}}] \leftrightarrow A^{*}\\
&\vdash x\,\underline{\mathtt{rnt}}\,A \to A^{*}
\end{aligned}
$$

and we find that soundness holds. An interesting corollary is

7.6. PROPOSITION. $\mathbf{HAS}^{*}$ *is closed under the* Uniformity Rule

UR $\qquad \vdash \forall X \exists y\, A(X,y) \;\Leftrightarrow \vdash \exists y \forall X\, A(X,y)$

and satisfies DP *and* EDN.

7.7. *Second-order extensions of other types of realizability*

The preceding two examples reveal something of a pattern for the extension to second-order languages. The pattern will become still clearer when we study the extension to higher-order logic in the next section, but let us already now indicate what has to be done to extend extensional and modified realizability.

In the case of extensional realizability, set variables should get assigned variables ranging over partial equivalence relations over $\mathbb{N}$. (A partial equivalence relation satisfies symmetry and transitivity, but not necessarily reflexivity). Second-order quantification is treated in the "uniform" way, just as for ordinary realizability.

In the case of modified realizability, there is no immediate generalization of the abstract version for $\mathbf{HA}^{\omega}$, but we can generalize HRO-$\underline{\mathtt{mr}}$-realizability; we shall abbreviate this as $\underline{\mathtt{mrn}}$-realizability ("modified realizability for numbers").

In this case we need to assign to each formula not only a set of realizers, but also a set of "potential realizers", which determine the domain of definition in the case of implication. (In the case of HRO-$\underline{\mathtt{mrn}}$-realizability restricted to $\mathbf{HA}^{\omega}$, the sets of potential realizers are always of the form HRO_{σ}.) In particular, we must assign to set variable X two variables X^{r} (representing the realizing numbers) and X^{d} (representing the set of potential realizers). We then define for each formula A the predicates

$x\,\underline{\texttt{mrn}}\,A$ ("x HRO-modified realizes A") and A^{d} (the set of potential realizers). Some typical clauses for the potential realizers: $(t = s)^{\mathrm{d}} := \mathbb{N}$, $(A \to B)^{\mathrm{d}} := \{x : \forall y{\in}A^{\mathrm{d}}(x{\bullet}y \in B^{\mathrm{d}})\}$, $(\forall X.A)^{\mathrm{d}} := \forall X^{\mathrm{d}}A^{\mathrm{d}}$, and for the realizability $x\,\underline{\texttt{mrn}}\,Xt := x \in X^{\mathrm{d}} \wedge X^{\mathrm{r}}(x,t)$, $x\,\underline{\texttt{mrn}}\,(A \to B) := x \in (A \to B)^{\mathrm{d}} \wedge \forall y(y\,\underline{\texttt{mrn}}\,A \to x{\bullet}y\,\underline{\texttt{mrn}}\,B)$, $x\,\underline{\texttt{mrn}}\,\forall X.A := \forall X^{\mathrm{d}}X^{\mathrm{r}}(x\,\underline{\texttt{mrn}}\,A)$.

The reader will have no difficulty in supplying the remaining ones, keeping in mind that this is to be an extension of HRO-$\underline{\texttt{mr}}$-realizability. However, in verifying soundness, it turns out that there is on important extra property required of the A^{d}: there should always be a fixed number in the sets of potential realizers, so that operations defined over the A^{d} must be defined at least somewhere. If we let the variables X^{d} range over inhabited sets containing 0, and if we choose our gödelnumbering of partial recursive operations in such a way that $\mathbf{p}(0,0) = 0$ and $\Lambda x.0 = 0$, it follows that $0 \in A^{\mathrm{d}}$ for all A.

7.8. *Realizability as a truth-value semantics*

It is instructive to rewrite $\underline{\texttt{rn}}$-realizability for $\mathbf{HA}^*$ in the form of a valuation in a set of truth-values. Let $X, Y \in \mathcal{P}(\mathbb{N})$; we define

Definition.

$$\begin{aligned} X \wedge Y &:= \{\mathbf{p}(x,y) : x \in X \wedge y \in Y\},\\ X \to Y &:= \{z : \forall x \in X(z{\bullet}x \in Y)\},\\ X \vee Y &:= \{\mathbf{p}(0,x) : x \in X\} \cup \{\mathbf{p}(Sz,y) : z \in \mathbb{N}, y \in Y\},\\ X \leftrightarrow Y &:= (X \to Y) \wedge (Y \to X). \end{aligned}$$

We associate to each formula A of $\mathbf{HA}^*$ a set $[\![A]\!]$ of realizing numbers:

$$\begin{aligned} [\![t = s]\!] &:= \{x : t = s\},\\ [\![A \wedge B]\!] &:= [\![A]\!] \wedge [\![B]\!],\\ [\![A \to B]\!] &:= [\![A]\!] \to [\![B]\!],\\ [\![\forall x A]\!] &:= \{z : \forall x(z{\bullet}x \in [\![A]\!])\},\\ [\![\exists x A]\!] &:= \{\mathbf{p}(y,z) : z \in [\![A[x/y]]\!]\}. \end{aligned}$$

The defined set contains the free variables of A as parameters. Furthermore we can put, in keeping with our definition of disjunction,

$$[\![A \vee B]\!] := [\![A]\!] \vee [\![B]\!]. \quad \square$$

The elements of $\mathcal{P}(\mathbb{N})$ act as truth-values; all inhabited elements represent "truth" in the sense of realizability.

If we now want to extend this to $\mathbf{HAS}^*$, we should put

$$\begin{aligned} [\![Xt]\!] &:= \{x : X^*(x,t)\},\\ [\![\forall X.A]\!] &:= \bigcap_X [\![A(X)]\!]. \end{aligned}$$

and now $[\![A]\!]$ contains for any X free in A a parameter X^*. (We may do without an explicit definition for the cases for $\wedge, \exists$ since these are definable in a second-order setting, cf. 7.3.)

Note that numerical and set quantifiers are treated in a completely different way. This can be remedied in this case in a more or less ad hoc manner: we associate with each domain D a set-valued function E_D on the elements, giving their "extent". In the case of **HAS*** we take

$$E_{\mathbb{N}}(n) := \{n\}, \quad E_{\mathcal{P}(\mathbb{N})}(X) := \{0\},$$

and define for domains D

$$[\![\forall x \in D.A(x)]\!] := \bigcap_{x'} (Ex' \to [\![A(x)]\!])$$

where x' is the parameter in $[\![A(x)]\!]$ corresponding to x (i.e. $x \equiv x'$ for numerical x, $x' = X^*$ if x' is a set variable X). (To see that the resulting notion of realizability is equivalent in the sense of 1.2, take for the ϕ and ψ: $\phi_{\forall xA}(y) := \Lambda x.\phi_{A(x)}(y\bullet x)$, $\psi_{\forall xA}(y) := \lambda x.\psi_{A(x)}(y\bullet x)$, $\phi_{\forall X.A}(y) := \lambda z.\phi_A(y)$ (z not free in $\phi_A(y)$), $\psi_{\forall X.A}(y) := \psi_A(y\bullet 0)$.) Such an ad hoc solution to enforce uniformity of definition will not be satisfactory in the case of higher-order logic, to be discussed in the next section.

7.9. *Notes*

(**?**) extended mrn-realizability, and (**?**) extended q-realizability to **HAS**; here we have recast Friedman'sFriedman, H.M. definition as rnt-realizability.

The idea of realizability as a truth-value semantics occurred to several researchers independently, shortly before 1980. The first documented reference to "realizability treated as a truth-value semantics" I could find is (**?**), cf. also (**?**). Other authors credit W. PowellPowell, W., or D.S. ScottScott, D. S. with the idea.

8 Realizability for higher-order logic and arithmetic

8.1. *Formulation of* **HAH**

Higher-order logic is based on a many-sorted language with a collection of *sorts* or *types*; we use $\sigma, \sigma', \ldots, \tau, \tau', \ldots$ for arbitrary types. There are variables $(x^\sigma, y^\sigma, z^\sigma, \ldots)$ for each type, and an equality symbol $=_\sigma$ for each σ. Relation symbols and function symbols may take arguments of different types. For quantifiers ranging over objects of type σ we sometimes write $\forall x{\in}\sigma$, $\exists x{\in}\sigma$ instead of $\forall x^\sigma$, $\exists x^\sigma$.

For intuitionistic and classical *higher-order logic* there are certain type-forming operations generating new types with appropriate axioms connecting the types.

Definition. (*Axioms and language for higher-order logic*) In a many-sorted language for higher-order logic, the collection of types is closed under $\times, \mathrm{P}, \to$, i.e.

(i) with each type σ there is a *power type* $\mathrm{P}(\sigma)$;

(ii) with each pair of types σ, τ there is a *product type* $\sigma \times \tau$ and a *function type* $\sigma \to \tau$.

One often includes a type ω of truth-values; then $\mathrm{P}(\sigma)$ may be identified with $\sigma \to \omega$.

There is a binary relation $\in_\sigma$ with arguments of type $\sigma, \mathrm{P}(\sigma)$; instead of $\in_\sigma(x, y)$ we write $x \in_\sigma y$ and sometimes $y(x)$ (predicate applied to argument).

For types $\sigma \to \tau, \sigma$ there is an application operation $\mathrm{App}_{\sigma,\tau}$ such that for $t \in \sigma{\to}\tau$, $t' \in \sigma$, $\mathrm{App}_{\sigma,\tau}(t, t')$ is a term of type τ. Usually we write tt' for $\mathrm{App}(t, t')$.

For each pair σ, τ there are functional constants $\mathbf{p}^{\sigma,\tau}, \mathbf{p}_0^{\sigma,\tau}, \mathbf{p}_1^{\sigma,\tau}$ such that $\mathbf{p}$ takes arguments of type σ, τ and yields a value of type $\sigma \times \tau$, $\mathbf{p}_0, \mathbf{p}_1$ take arguments of type $\sigma \times \tau$ and yield values of type σ and τ respectively. The pairing axioms are assumed:

PAIR $\quad \forall x_0 x_1 (\mathbf{p}_i(\mathbf{p}(x_0, x_1) = x_i) \quad (i = 0, 1)$

SURJ $\quad \forall x^{\sigma\times\tau}(\mathbf{p}(\mathbf{p}_0 x, \mathbf{p}_1 x) = x).$

For power-types we require replacement

REPL $\quad \forall X^{\mathrm{P}(\sigma)} \forall x^\sigma y^\sigma (x \in X \wedge x = y \to y \in X),$

as well as extensionality and comprehension:

EXT $\quad \forall X^{\mathrm{P}(\sigma)} Y^{\mathrm{P}(\sigma)} (\forall x^\sigma (x \in X \leftrightarrow x \in Y) \to X = Y)),$

CA $\quad \exists X^{\mathrm{P}(\sigma)} \forall x^\sigma (x \in X \leftrightarrow A(x)).$

For function types the corresponding requirements are

EXTF $\quad \forall y^{\sigma\to\tau} z^{\sigma\to\tau} (\forall x^\sigma (yx = zx) \to y = z)$

CAF $\quad \forall x^\sigma \exists! y^\tau A(x, y) \to \exists z^{\sigma\to\tau} \forall x^\sigma A(x, zx).$

If the type ω is present and $\mathrm{P}(\sigma)$ is identified with $\sigma \to \omega$, EXT and CA become special cases of EXTF and CAF, and REPL follows from the fact that functions respect equality.

8.2. Definition. **HAH**, *intuitionistic higher-order arithmetic* ("HeytingHeyting, A. Arithmetic of Higher order") is a specialization of higher-order logic based on a single basic type 0 (or N) for the natural numbers; types are closed under power-type and function-type formation.

On the basis type 0 an injective function $S : 0 \to 0$ is given, with axioms $Sx = Sy \to x = y$, $0 \neq Sx$. Defining

$$x \in \mathbb{N} := \forall X(0 \in X \wedge \forall y(Xy \to X(Sy)) \to x \in X)$$

we add an axiom stating that all elements of type 0 are in $\mathbb{N}$: $\forall x^0(x \in \mathbb{N})$. As a result, the induction axiom becomes valid. □

Remarks. (i) **E-HA**$^\omega$ is a fragment of **HAH** based on type 0 and function-type formation only.

(ii) It is well known, that if we consider in **HAH** any set X with a special element $x_0 \in X$ and a function $f : X \to X$, then there is a unique function $F : \mathbb{N} \to X$ such that $F0 = x_0$, $F(Sx) = f(Fx)$. In particular, if f is injective, then the image $f[X] \cup \{x_0\}$ is isomorphic to the type N.

8.3. *Numerical realizability for many-sorted logic*

Since our versions of intuitionistic higher-order logic, and the system **HAH** are based on intuitionistic many-sorted predicate logic, we first discuss realizability for many-sorted logic. Our definition of realizability will be motivated by the truth-functional reformulation of realizability for **HAS** in 7.8.

We start with realizability for many-sorted logic without function symbols. Below $\Omega \equiv \mathcal{P}(\mathbb{N})$, Ω^* is the collection of all inhabited subsets of $\mathbb{N}$. We first introduce Ω-sets, which will serve to interpret the types with their equalities.

8.4. Definition. An Ω-*set* $\mathcal{X} \equiv (X, =_\mathcal{X})$ is a set X together with a map $=_\mathcal{X}$: $X^2 \longrightarrow \Omega$ such that the following is true (writing $t =_\mathcal{X} t'$ for $=_\mathcal{X}(t, t')$):

$$\bigcap_{x,y}(x =_\mathcal{X} y \to y =_\mathcal{X} x) \in \Omega^*,$$
$$\bigcap_{x,y,z}(x =_\mathcal{X} y \wedge y =_\mathcal{X} z \to x =_\mathcal{X} z) \in \Omega^*.$$

Here $\wedge, \to$ on the left have to be understood as defined for elements of Ω, as in 7.8. We write $E_\mathcal{X} t$ for $t =_\mathcal{X} t$.5

The Ω-*product* of two Ω-sets $\mathcal{X} \equiv (X, \sim)$ and $\mathcal{Y} \equiv (Y, \sim')$ is the Ω-set $\mathcal{X} \times \mathcal{Y} \equiv (X \times Y, \sim'')$ where

$$(x, y) \sim'' (x', y') := (x \sim x') \wedge (y \sim' y').$$

A product of n factors $\mathcal{X}_1, \ldots, \mathcal{X}_n$ is defined as $(\mathcal{X}_1 \times \cdots \times X_{n-1}) \times \mathcal{X}_n$.

We use calligraphic capitals $\mathcal{X}, \mathcal{Y}, \ldots$ for Ω-sets. □

EXAMPLES. Ω itself may be viewed as an Ω-set $(\Omega, \leftrightarrow)$ where $X \leftrightarrow Y$ is defined as in 7.8. Another example is $\mathcal{N} := (\mathbb{N}, =_{\mathbb{N}})$, where $n =_{\mathbb{N}} m := \{n\} \cap \{m\} \equiv \{n : n = m\}$.

8.5. DEFINITION. Let $\mathcal{X} \equiv (X, \sim)$ be an Ω-set and $F : X \longrightarrow \Omega$ a map. We put

$$\begin{aligned} \mathrm{Strict}(F) &:= \bigcap_{x \in X} (Fx \to Ex), \\ \mathrm{Repl}(F) &:= \bigcap_{x,y \in X} (Fx \wedge x \sim y \to Fy). \end{aligned}$$

An Ω-*predicate* on $\mathcal{X}$ is an $F : X \to \Omega$ such that Strict(F) and Repl(F) are inhabited (belong to Ω^*). An Ω-*relation* on $\mathcal{X}_1, \ldots, \mathcal{X}_n$ is an Ω-predicate on $\mathcal{X}_1 \times \cdots \times \mathcal{X}_n$.

If $(X \times Y, \sim)$ is the product of the Ω-sets $(X, =_X)$ and $(Y, =_Y)$, and $F : X \times Y \longrightarrow \Omega$, we define

$$\begin{aligned} \mathrm{Fun}(F) &:= \bigcap_{x,y,z} (F(x,y) \wedge F(x,z) \to y =_Y z) \\ \mathrm{Total}(F) &:= \bigcap_{x} (Ex \to \bigcup_{y} F(x,y)). \end{aligned}$$

An Ω-*function* from $\mathcal{X}$ to $\mathcal{Y}$ is an $F : X \times Y \to \Omega$ such that Strict(F), Repl(F), Fun(F), Total(F) are inhabited. The definition of Ω-function for more than one argument is reduced to this case via products of Ω-sets. $\square$

8.6. DEFINITION. An *interpretation* $[\![\ ]\!]$ of a many-sorted relational language assigns

(i) to each type σ with equality $=_\sigma$ an Ω-set $[\![\sigma]\!] \equiv (\bar{\sigma}, [\![=_\sigma]\!])$; for $[\![=_\sigma]\!](x,x)$ we also write $E_\sigma x$, and we shall permit ourselves in the sequel a slight abuse of language, using $[\![\sigma]\!]$ also for the underlying set $\bar{\sigma}$.

(ii) to constants c of type σ an element $[\![c]\!]$ of $[\![\sigma]\!]$,

(iii) to each n-ary relation symbol R, taking arguments of sorts $\sigma_1, \ldots, \sigma_n$ respectively, an Ω-relation $[\![R]\!]$ on $[\![\sigma_1]\!], \ldots, [\![\sigma_n]\!]$. $\square$

N.B. It is important to observe that for practical purposes the definition of Ex for an Ω-set $(X, \sim)$ may be liberalized, it suffices that $\bigcap_x (Ex \leftrightarrow x \sim x) \in \Omega^*$.

REMARK. If in the definition above we take for Ω a complete HeytingHeyting, A. algebra, and replace $\bigcap$ in the conditions above by the meet operator $\bigwedge$, and take $\Omega^* := \{\top\}$, $\top$ the top element of Ω, we obtain precisely the interpretation of many-sorted intuitionistic logic in Ω-sets, as described in (**?**) or (**?**). There Et measures the "degree of existence" of t.

8.7. DEFINITION. Let $\mathcal{X} \equiv (X, \sim)$ be an Ω-set, and let $F : X \to \Omega$, then

$$\forall x {\in} \mathcal{X} F(x) := \bigcap_{x \in X} (E_{\mathcal{X}} x \to Fx), \quad \exists x {\in} \mathcal{X} F(x) := \bigcup_{x \in X} (E_{\mathcal{X}} x \wedge Fx). \quad \square$$

N.B. "$\forall x \in \mathcal{X} \ldots$", "$\exists x \in \mathcal{X} \ldots$" indicate elements of Ω, but "$\forall x \in X \ldots$", "$\exists x \in X \ldots$" refer to ordinary quantification.

8.8. DEFINITION. The *interpretation* of *formulas* of a many-sorted relational language may now be given modulo assignments ρ for the variables. Let ρ be an assignment of elements of $[\![\sigma]\!]$ to the variables of type σ, for all σ. For constants c the interpretation $[\![c]\!]_\rho$ is supposed to be given; for variables $[\![x]\!]_\rho := \rho(x)$, and for prime formulas

$$[\![t =_\sigma t']\!]_\rho := [\![=_\sigma]\!]([\![t]\!]_\rho, [\![t']\!]_\rho), \quad [\![R(t_1, \ldots, t_n)]\!]_\rho := [\![R]\!]([\![t_1]\!]_\rho, \ldots, [\![t_n]\!]_\rho),$$

and for compound formulas according to 7.8, i.e.

$$\begin{array}{ll} [\![A \wedge B]\!]_\rho & := [\![A]\!]_\rho \wedge [\![B]\!]_\rho, \\ [\![A \to B]\!]_\rho & := [\![A]\!]_\rho \to [\![B]\!]_\rho, \\ [\![\neg A]\!]_\rho & := [\![A]\!]_\rho \to \emptyset, \\ [\![\forall x {\in} \sigma . A]\!]_\rho & := (\forall d {\in} [\![\sigma]\!]) [\![A]\!]_{\rho[x/d]}, \\ [\![\exists x {\in} \sigma . A]\!]_\rho & := (\exists d {\in} [\![\sigma]\!]) [\![A]\!]_{\rho[x/d]}, \end{array}$$

where $\rho[x/d]$ is the assignment given by $\rho[x/d](y) = \rho(y)$ for $y \not\equiv x$, $\rho[x/d](x) = d$. Instead of using assignments, we may also use a language enriched with constants as names for each Ω-set used for the interpretation of the types, and define the interpretation only for sentences.

A sentence A is said to be *valid* if $[\![A]\!] \in \Omega^*$. $\square$

N.B. In the sequel we shall sometimes used "mixed" expressions: for an Ω-set $\mathcal{X} \equiv (X, \sim)$ $[\![\forall x \in \mathcal{X} A(x)]\!] := \bigcap_{x \in X} (Ex \to [\![A(x)]\!])$, $[\![\exists x \in \mathcal{X} A(x)]\!] := \bigcup_{x \in X} (Ex \wedge [\![A(x)]\!])$.

8.9. PROPOSITION. *Intuitionistic many-sorted predicate logic is sound for realizability.*

Proof. The proof is routine. The definition of an interpretation says that for the Ω-relation $[\![R]\!]$ interpreting relation R of the language the following hold:

(1) $$\bigcap_{x,y} ([\![x = y]\!] \to [\![y = x]\!]) \in \Omega^*,$$

(2) $$\bigcap_{x,y,z} ([\![x = y]\!] \wedge [\![y = z]\!] \to [\![x = z]\!]) \in \Omega^*,$$

(3) $$\bigcap_{x_1, \ldots, x_n} ([\![R]\!](x_1, \ldots, x_n) \to Ex_1 \wedge \ldots \wedge Ex_n) \in \Omega^*,$$

(4) $$\bigcap_{\vec{x},\vec{y}}(\llbracket R\rrbracket(\vec{x}) \wedge \llbracket \vec{x}=\vec{y}\rrbracket \to \llbracket R\rrbracket(\vec{y})) \in \Omega^*$$

where of course $\llbracket \vec{x}=\vec{y}\rrbracket$ abbreviates $\llbracket x_1 = y_1\rrbracket \wedge \ \ldots \wedge \llbracket x_n = y_n\rrbracket$; similarly we may abbreviate $Ex_1 \wedge \ldots \wedge E_n$ as $E\vec{x}$.

(1) and (2) guarantee the validity of symmetry and transitivity of equality, and (3) and (4) the validity of strictness and replacement for R. Reflexivity translates into the trivial $\bigcap_x(Ex \to \llbracket x=x\rrbracket) \in \Omega^*$, so does not need an extra condition. □

The definition of the interpretation of a language with function symbols is reduced to the case of relational languages, by regarding functions as special relations (a partial function is a relation which is functional: $\forall\vec{x}yz(R(\vec{x},y) \wedge R(\vec{x},z) \to y=z)$, and a (total) function is a relation which is functional and total, i.e. satisfies $\forall\vec{x}\exists yR(\vec{x},y)$).

8.10. Definition. (*Interpretation of function symbols*) To each function symbol $F : \sigma_1 \times \cdots \times \sigma_n \to \sigma$ we assign an Ω-function $\llbracket F\rrbracket$ from $\llbracket\sigma_1\rrbracket \times \cdots \times \llbracket\sigma_n\rrbracket$ to $\llbracket\sigma\rrbracket$. In full the conditions read:

$$\bigcap_{\vec{x}}(E\vec{x} \to \textstyle\bigcup_y \llbracket F\rrbracket(\vec{x},y)) \in \Omega^*,$$
$$\bigcap_{\vec{x},y,z}(\llbracket F\rrbracket(\vec{x},y) \wedge \llbracket F\rrbracket(\vec{x},z) \to \llbracket y=z\rrbracket) \in \Omega^*.$$

For partial functions the first condition may be omitted. □
As to the reason for using relations to interpret functions, see 8.11.

8.11. Definition. (*Interpretation of formulas for languages with function symbols*) We now assume a language with relation symbols and symbols for total functions.

We have to say how to interpret $t_1 = t_2$ and $Rt_1 \ldots t_n$ for compound terms $t_1, \ldots, t_n$. This is done recursively: $t_1 = t_2$ for arbitrary t_1, t_2 is interpreted as $\exists x(t_1 = x \wedge t_2 = x)$; $Ft_1 \ldots t_n = x$ is interpreted as $\exists x_1 \ldots x_n(t_1 = x_1 \wedge \ldots \wedge t_n = x_n \wedge Fx_1 \ldots x_n = x)$; the value of $Fx_1 \ldots x_n = x$ is given by $\llbracket F\rrbracket(\rho x_1, \ldots, \rho x_n, \rho x)$. $Rt_1 \ldots t_n$ is interpreted as $\exists\vec{x}(\vec{t} = \vec{x} \wedge R(\vec{x}))$. □

8.12. Theorem. *The interpretation above is sound for many-sorted logic.*
Proof. Almost entirely routine. To see that e.g. all instances of $\forall xA \to A[x/t]$ are valid, one should note that the "unwinding" of $t_1 = t_2$ and $Rt_1 \ldots t_n$ mentioned above is precisely what one does in showing syntactically that the addition of symbols for definable functions with the appropriate axiom is conservative; the standard proof, e.g. (**?**), shows that the "unwinding" translation of $\forall xA \to A[x/t]$ is in fact derivable in the relational part of the language.

In our case this means that the soundness reduces to the soundness for a relational language with an extra relation symbol R_F for each function symbol F in the original language. □.

8.13. Remark. The reason that we have not imposed the stronger requirement that a function symbol $F : \sigma_1 \times \cdots \times \sigma_n \to \sigma$ is to be interpreted by a function $[\![F]\!] : [\![\sigma_1]\!] \times \cdots \times [\![\sigma_n]\!] \to [\![\sigma]\!]$ lies in the fact that this sometimes not sufficiently general: the interpretation of $\forall x \exists! y R(x, y)$ says that $\bigcap_x (Ex \to \bigcup_y [\![R]\!](x, y)) \in \Omega^*$, and $\bigcap_{x,y,z}([\![R]\!](x, y) \wedge [\![R]\!](x, z) \to [\![y = z]\!]) \in \Omega^*$, but there is no guarantee that we can find a function f such that $\bigcap_x [\![R]\!](x, fx) \in \Omega^*$. Cf. the similar situation for the interpretations where Ω is a complete HeytingHeyting, A. algebra; only for sheaves the situation simplifies; see (**?**, 1.3.16, chapter 14).

Example. Let f be a primitive recursive function with function symbol F in the language of **HA**. Over the domain $\mathcal{N} \equiv (\mathbb{N}, [\![=_{\mathbb{N}}]\!])$ as defined above we can introduce the interpretation of F by the relation

$$R_F(n, m) := En \wedge [\![m = fn]\!]$$

we add En to guarantee the strictness of R_F), so $R_F := \{\mathbf{p}(n, fn) : n \in \mathbb{N}\}$. It is now routine to see that this yields a realizability for **HA** (equivalent to) the one defined before.

8.14. Remark. The modelling of many-sorted logic described above is an interpretation in a certain category Eff, with as objects the Ω-sets, and as morphisms (equivalence classes of) Ω-functions. More precisely, the morphisms from $\mathcal{X}$ to $\mathcal{Y}$ are given by Ω-relations on $\mathcal{X} \times \mathcal{Y}$ such that (cf. definition **??**)

$$\text{Strict}(F),\ \text{Repl}(F),\ \text{Fun}(F),\ \text{Total}(F) \in \Omega^*,$$

modulo an equivalence $\approx$ defined as

$$F \approx F' := [\![\forall xy(Fxy \leftrightarrow F'xy)]\!] = \bigcap_{x,y} (Ex \wedge Ey \to (Fxy \leftrightarrow F'xy)) \in \Omega^*.$$

Composition of morphisms $F : \mathcal{X} \to \mathcal{Y}$, $G : \mathcal{Y} \to \mathcal{Z}$ is given by the relational product: $G \circ F$ is the relation on $\mathcal{X} \times \mathcal{Z}$ given by

$$(G \circ F)(x, z) := \exists y \in \mathcal{Y}(F(x, y) \wedge G(y, z)).$$

The identity morphism $\text{id}_{\mathcal{X}} : \mathcal{X} \to \mathcal{X}$ is simply (the equivalence class of) $\text{id}_{\mathcal{X}}(x, y) := x =_{\mathcal{X}} y$.

Let Sets be the category of sets and set-theoretic mappings. There are functors $\Delta : \mathsf{Sets} \to \mathsf{Eff}$ and $\Gamma : \mathsf{Eff} \to \mathsf{Sets}$ which may be described as follows.

Δ, the "*constant-objects functor*", maps a set X to the Ω-set $(X,\sim)$ with $x \sim y := \{n \in \mathbb{N} : x = y\}$, and if $f : X \to Y$, then Δf is represented by the Ω-relation R_f, $R_f := (fx \sim y)$.

$\Gamma(X,\sim) = \{x : Ex \text{ inhabited}\}/{\simeq}$, where $\simeq$ is the equivalence relation $x \simeq x' := (x \sim x' \text{ inhabited})$. Γ is usually called the *global-sections functor*, since it is naturally isomorphic tot the functor which assigns to $(X,\sim)$ the set of morphisms $\top \to (X,\sim)$; $\top$ is the terminal object $(\{*\},=_*)$ with $(* =_* *) = \mathbb{N}$.

Δ preserves finite limits and is full and faithful; Γ also preserves finite limits, and Γ is left-adjoint to Δ. $\mathcal{N}$ is a natural-numbers object in Eff, and Eff is in fact a topos (see 8.15). For the theory of Eff, with proofs of the facts mentioned above, see (**?**); the general theory of realizability toposes is treated in (**?**). Other sources of information on Eff are (**?**) and (**?**).

The categorical view provided by Eff suggests the following

8.15. Definition. The Ω-sets $\mathcal{X} \equiv (X,\sim)$, $\mathcal{Y} \equiv (Y,\sim')$ are said to be *isomorphic* if there are Ω-functions $F : X \times Y \to \Omega$, $G : Y \times X \to \Omega$ such that $G \circ F \approx \mathrm{id}_{\mathcal{X}}$, $F \circ G \approx \mathrm{id}_{\mathcal{Y}}$, where $\approx$ is defined as above, i.e., $H \approx H' := [\![\forall xy(Hxy \leftrightarrow H'xy)]\!]$. $\square$

It is easy to see that quantification over isomorphic sets yields equivalent results, in the following sense:

Lemma. *Let $\mathcal{X}$ and $\mathcal{Y}$ be isomorphic via F, G. Then*

$$\forall x{\in}\mathcal{X}.A(x) = \forall y{\in}\mathcal{Y}.A(Gy) = \forall y{\in}\mathcal{Y}\forall x{\in}\mathcal{X}.(G(y,x) \to A(x)),$$

and similarly for existential quantification.

The proof is routine, relying on the soundness of logic. Important special cases are (a) any Ω-set $\mathcal{X} \equiv (X,\sim)$ is isomorphic to $(X', \sim\restriction X' \times X')$ where $X' := \{x : x \in X \text{ and } Ex \in \Omega^*\}$, and (b) the following situation: let $\mathcal{X} \equiv (X,\sim)$ be an Ω-set, and let $X' \subset X$ such that

$$\forall x \in X(Ex \in \Omega^* \to \exists x' \in X'(x \sim x') \in \Omega^*).$$

8.16. *Products, powersets and exponentials*

Definition. The interpretation of a product $[\![\sigma_1 \times \sigma_2]\!]$ is $[\![\sigma_1]\!] \times [\![\sigma_2]\!]$. The functions $\mathbf{p}^{\sigma,\tau}, \mathbf{p}_0^{\sigma,\tau}, \mathbf{p}_1^{\sigma,\tau}$ are simply represented by the pairing and unpairing on the relevant Ω-sets. $\square$

In order to interpret higher-order logic, the interpretation of type σ and types $P(\sigma)$ and $\sigma \to \tau$ relative to the interpretation of σ, τ, and the interpretation of the relation $\in_\sigma$ as well as the operator App must be such that extensionality and comprehension are valid.

DEFINITION. Let $\mathcal{X} \equiv (X, \sim)$ be an Ω-set; the Ω-*powerset* of $\mathcal{X}$, $\mathrm{P}(\mathcal{X})$, is an Ω-set $(X \to \Omega, \simeq)$ where for F, G of $X \to \Omega$:

$$\begin{aligned} E(F) &:= \mathrm{Strict}(F) \wedge \mathrm{Repl}(F), \\ F \simeq G &:= E(F) \wedge E(G) \wedge \textstyle\bigcap_x (Fx \leftrightarrow Gx), \\ x \in_{\mathcal{X}} F &:= Fx \wedge EF. \end{aligned}$$

Let $\mathcal{X} \equiv (X, \sim), \mathcal{Y} \equiv (Y, \sim')$ be Ω-sets, then the Ω-*functionset* $\mathcal{X} \to \mathcal{Y}$ is $(X \times Y \to \Omega, \approx)$ such that for $F, G \in X \times Y \to \Omega$

$$\begin{aligned} E(F) &:= \mathrm{Str}(F) \wedge \mathrm{Repl}(F) \wedge \mathrm{Fun}(F) \wedge \mathrm{Total}(F) \\ F \approx G &:= \textstyle\bigcap_{x,y} (Ex \wedge Ey \to (Fxy \leftrightarrow Gxy)) \wedge EF \wedge EG, \\ \mathrm{App}_{\mathcal{X},\mathcal{Y}}(F, x, y) &:= EF \wedge Fxy. \end{aligned}$$

N.B. Here we have availed ourselves of the freedom to define $E(F)$ so as to be equivalent only to $F \approx F$ in the realizability sense, not literally identical. $\square$

REMARK. As noted above, Ω itself may be viewed as an Ω-set $(\Omega, \sim)$. It is then not hard to see that the Ω-powerset of a Ω-set $\mathcal{X}$ is in fact isomorphic to the Ω-functionset $\mathcal{X} \to \Omega$.

DEFINITION. In an interpretation of intuitionistic higher-order logic powertypes and exponentials are interpreted such that $[\![\mathrm{P}(\sigma)]\!]$ is the $\mathrm{P}[\![\sigma]\!]$ and such that $[\![\sigma \to \tau]\!]$ is $[\![\sigma]\!] \to [\![\tau]\!]$. $[\![\in_\sigma]\!] := \in_{[\![\sigma]\!]}$, $[\![\mathrm{App}_{\sigma,\tau}]\!] := \mathrm{App}_{[\![\sigma]\!],[\![\tau]\!]}$. $\square$

REMARK. We can easily introduce subtypes of a given type, as follows. Let $\mathcal{X} \equiv (X, \sim)$ be an Ω-set. Intuitively an Ω-predicate over $\mathcal{X}$ determines a subset of $\mathcal{X}$. We may again make this into an Ω-set:

DEFINITION. Let $\mathcal{X} \equiv (X, \sim)$ be an Ω-set, and let $F : X \to \Omega$ be an Ω-predicate; then the Ω-subset of $\mathcal{X}$ determined by F is $(X', \sim')$ with $X' := \{x : Fx \in \Omega^*\}$, $x \sim' y := x \sim y$. $\square$
N.B. An equivalent definition would have been obtained by taking $X' = X$, and $x \sim y := Fx \wedge Fy \wedge x \sim y$. The resulting Ω-set is isomorphic to the one defined above.

8.17. PROPOSITION. *Extensionality and comprehension are valid.*
The proof is routine.

REMARKS. (i) In categorical terms, the preceding facts mean that the category Eff has products and exponentials (i.e. is cartesian closed), and moreover has a classifying truth-value object, namely $(\Omega, \leftrightarrow)$, and hence is a topos.

The fact that the natural numbers are unique modulo isomorphism in higher-order logic (8.2) corresponds in categorical terms to the uniqueness of the natural number object in a topos.

(ii) Obviously, the notions needed for the realizability interpretation of **HAH** can be formalized in **HAH** itself. If we assign a *level* $\ell(\sigma)$ to types σ according to $\ell(0) = \ell(\omega) = 0$, $\ell(\mathrm{P}\sigma) = \ell(\sigma)+1$, $\ell(\sigma{\to}\tau) = \max(\ell(\sigma)+1, \ell(\tau)$, $\ell(\sigma{\times}\tau) = \max(\ell(\sigma), \ell(\tau))$, then the interpretation of a formula of level $\leq n$ (i.e. all variables are of level $\leq n$) is definable by a formula of level $\leq n$.

Our next aim will be to show that for **HAS** the resulting notion of realizability is in fact equivalent to realizability as defined in 7.2.

8.18. Definition. An Ω-set $\mathcal{X} \equiv (X, \sim)$ is called *canonically uniform* if $\bigcap_{x\in X} Ex$ is inhabited. $\mathcal{X}$ is *uniform* if it is isomorphic to a canonically uniform set.

8.19. Lemma. *For uniform Ω-sets $\mathcal{X} \equiv (X, \sim)$ interpretation of universal and existential quantifiers may be simplified to*

$$\forall x{\in}\mathcal{X}\, Fx := \bigcap_{x\in X} Fx, \quad \exists x{\in}\mathcal{X}\, Fx := \bigcup_{x\in X} Fx;$$

more precisely, $(\bigcap_x (Ex \to Fx) \leftrightarrow \bigcap_x Fx) \in \Omega^*$, $(\bigcup_x (Ex \wedge Fx) \leftrightarrow \bigcup_x Fx) \in \Omega^*$.

proof. It suffices to prove this for canonically uniform Ω-sets, and then it is easy: let $n \in \bigcap_x Ex$, and let $m \in \bigcap_x (Ex \to Fx)$, then $\forall x(m{\bullet}n \in Fx)$, i.e. $m{\bullet}n \in \bigcap_x Fx$, etc. $\square$

8.20. Definition. Let $\mathcal{X} \equiv (X, \sim)$ be an Ω-set. $\mathcal{X}$ is *canonically separated* if

$$(x \sim y) \text{ inhabited } \Rightarrow x = y.$$

$\mathcal{X}$ is *canonically proto-effective* if

$$Ex \cap Ey \text{ inhabited } \Rightarrow x = y.$$

$\mathcal{X}$ is *separated* [*proto-effective*] if $\mathcal{X}$ is isomorphic to a canonically separated [canonically proto-effective] Ω-set. $\mathcal{X}$ is [*canonically*] *effective* if $\mathcal{X}$ is [canonically] separated and [canonically] effective. $\square$

8.21. Proposition. *Let $\mathcal{X} \equiv (X, \sim)$ be a uniform and $\mathcal{Y} \equiv (Y, \sim')$ a proto-effective Ω-set. Then the uniformity principle*

$$\mathrm{UP}(\mathcal{X}, \mathcal{Y}) \qquad \forall x{\in}\mathcal{X}\, \exists y{\in}\mathcal{Y}\, A(x, y) \to \exists y{\in}\mathcal{Y}\, \forall x{\in}\mathcal{X}\, A(x, y)$$

is valid.

Proof. Without loss of generality we may assume $\mathcal{Y}$ to be canonically proto-effective. Let $n \in \bigcap_{x\in X} \exists y \in \mathcal{Y} A(x,y)$, so $n \in \bigcup_{y\in Y}(Ey \wedge A(x,y))$, then $\forall x{\in}X \exists y{\in}Y(\mathbf{p}_0 n \in Ey)$, i.e. $n \in \bigcup_{y\in Y} Ey \wedge \bigcap_{x\in X} A(x,y)$. □

The following proposition is not needed in what follows but describes the logical significance of separatedness:

8.22. Proposition. *An Ω-set $\mathcal{X} \equiv (X, \sim)$ is separated iff $\forall x, y{\in}\mathcal{X}(\neg\neg x{\sim}y \to x{\sim}y)$ is valid.*

Proof. It is easy to see that a canonically separated Ω-set $\mathcal{X}$ satisfies $\forall x, y{\in}\mathcal{X}(\neg\neg x{\sim}y \to x{\sim}y)$. Conversely, assume $\forall x, y{\in}\mathcal{X}(\neg\neg x{\sim}y \to x{\sim}y)$ to be valid. We define for $x, y \in X$:

$$\begin{aligned}
&[x] && := \{x' \in X \,:\, x' \sim x \text{ inhabited}\},\\
&[x] \approx [y] && := \{n \in \mathbb{N} \,:\, x \sim y \text{ inhabited}\},\\
&X' && := \{[x] \,:\, x \in X\}.
\end{aligned}$$

Then $\mathcal{X}' \equiv (X', \approx)$ is canonically separated and isomorphic to $\mathcal{X}$ via the Ω-relations F on $\mathcal{X} \times \mathcal{X}$ and G on $\mathcal{X} \times \mathcal{X}'$, defined by

$$F([x], y) \equiv G(x, [y]) := \{n \in \mathbb{N} \,:\, y \sim x \text{ inhabited}\}.$$

We have to show that F, G are strict, total, functional and that their composition is the identity. This is mostly routine. For example, to see that F is functional, observe that our hypothesis gives the existence of an n such that

$$\forall m, m'(m \in Ex \wedge m' \in Ey \wedge (\exists m''(m'' \in x{\sim}y) \to n{\bullet}\mathbf{p}(m, m') \in (x{\sim}y)). \tag{1}$$

The functionality of F amounts to validity of $\forall [x], y, y'(F([x], y) \wedge F([x], y') \to y \sim y')$, i.e.

$$\bigcap_{[x]\in X'} \bigcap_{y,y'\in X} (E[x] \wedge Ey \wedge Ey' \wedge \{n \in \mathbb{N} \,:\, \exists m(m \in x \sim y)\} \wedge \\ \{n{\in}\mathbb{N} \,:\, \exists m(m \in x \sim y')\} \to y \sim y') \in \Omega^*. \tag{2}$$

Since $x \sim y$ and $x \sim y'$ inhabited implies $y \sim y'$ inhabited, (2) readily follows from (1). □

The following proposition, with a proof due to van Oosten, justifies the terminology "uniform".

8.23. Proposition. *An Ω-set $\mathcal{X} \equiv (X, \sim)$ is uniform iff $\mathcal{X}$ satisfies* UP*$(\mathcal{X}, \mathcal{N})$.*

Proof. One direction is a consequence of proposition 8.19. The other direction is proved as follows. Given $\mathcal{X}$, consider the Ω-set $\mathcal{Y} \equiv (Y, \simeq)$ defined by

$$\begin{aligned}
&Y := \{\mathbf{p}(x, n) \,:\, n \in E_{\mathcal{X}} x\},\\
&(x, n) \simeq (y, m) := \{n \,:\, n \in x{\sim}y \wedge n = m\}.
\end{aligned}$$

We shall write (y, n) for $\mathbf{p}(y, n)$ in what follows. There is an Ω-function $G : \mathcal{Y} \to \mathcal{X}$ given by

$$G((x, n), x') := (x \sim x') \wedge \{n\}.$$

G is surjective as an Ω-function, i.e.

$$\forall y{\in}\mathcal{Y}\exists x{\in}\mathcal{X}.G(y, x),$$

i.e.

$$\bigcap_{(y,n)\in\mathcal{Y}} (E_{\mathcal{Y}}(y, n) \to \bigcup_{x\in X} (E_{\mathcal{X}}x \wedge G((y, n), x)) \in \Omega^*.$$

Let $H : \mathcal{Y} \to \mathcal{N}$ be the surjective Ω-function

$$H((x, n), m) := \{n : n = m\}.$$

Then

$$\forall x{\in}\mathcal{X}\, \exists y{\in}\mathcal{Y}\, \exists n{\in}\mathcal{N}(G(y, x) \wedge H(y, n))$$

is valid. If we assume $\mathrm{UP}(\mathcal{X}, \mathcal{N})$, it follows that

$$\exists n{\in}\mathcal{N}\, \forall x{\in}\mathcal{X}\, \exists y{\in}\mathcal{Y}(G(y, x) \wedge H(y, n))$$

is valid, which means that for some $n \in \mathbb{N}$

(1) $$\bigcap_{x\in X}(Ex \to \bigcup_{x'\in X, n\in Ex'} (x \sim x'))$$

is inhabited. Let now $\mathcal{Z} := (Z, \sim')$, $Z := \{x{\in}X : n{\in}Ex\}$, $\sim'$ the restriction of $\sim$ to Z. Clearly F defined by $F(x, x') := (x \sim x')$ is an injection of $\mathcal{Z}$ into $\mathcal{X}$; and the formula (1) states that this injection is also a surjection, hence $\mathcal{Z}$ and $\mathcal{X}$ are isomorphic. □

8.24. PROPOSITION. *The Ω-powerset of a separated Ω-set $\mathcal{X} = (X, \sim)$ is uniform.*

Proof. Let $\mathcal{X}$ be canonically separated, and let $\mathcal{Y} : X \to \Omega$ be an element of the Ω-powerset $\mathrm{P}(\mathcal{X})$ of $\mathcal{X}$, then $\Lambda k.\mathbf{p}_0 k$ realizes $\mathrm{Repl}(\mathcal{Y})$; $n \in \mathrm{Str}(\mathcal{Y})$ means $n \in \bigcap_{x\in X}(x \in \mathcal{Y} \to Ex)$.

By restricting attention to “normal” $\mathcal{Y}$ we can construct uniform realizers for Ey. Let us call $\mathcal{Y}$ *normal* if

$$m \in \mathcal{Y}(x) \Rightarrow \mathbf{p}_1 m \in Ex.$$

For normal $\mathcal{Y}$, $\Lambda m.\mathbf{p}_1 m \in \mathrm{Str}(\mathcal{Y})$, and so always

$$\mathbf{p}(\Lambda k.\mathbf{p}_0 k, \Lambda m.\mathbf{p}_1 m) \in E(\mathcal{Y}).$$

To show Ω-isomorphism of $\mathrm{P}(\mathcal{X})$ with the subset of normal elements, observe that if we map arbitrary $\mathcal{Y}$ to $\Phi(\mathcal{Y})$ with $\Phi(\mathcal{Y})(x) := \{\mathbf{p}(n, m) : n \in \mathcal{Y}(x) \wedge m \in Ex\}$, we have that $\mathcal{Y} =_{\mathrm{P}(\mathcal{X})} \Phi(\mathcal{Y})$ is inhabited:

$$\bigcap_{x\in X} (x \in \mathcal{Y} \leftrightarrow x \in \Phi(\mathcal{Y})) \in \Omega^*.$$

If $n \in \mathrm{Str}(\mathcal{Y})$, $k \in (x \in \mathcal{Y})$, then $n{\bullet}k \in Ex$, and $\mathbf{p}(k, n{\bullet}k) \in \Phi(\mathcal{Y})(x)$, etc. □

8.25. Proposition. *For* **HAS** *the realizability as defined above is equivalent to* $\underline{\texttt{rn}}$ *as defined in 7.2.*

Proof. This result is now obtainable as a corollary to the preceding propositions. As to the second-order quantifiers, we may restrict attention to the normal elements of the Ω-powerset of $\mathcal{N}$. Let

$$\begin{aligned}\Phi'(X) &:= \{\mathbf{p}(\mathbf{p}(x,y),y) : \mathbf{p}(x,y) \in X\},\\ \Phi''(X) &:= \{\mathbf{p}(x,y) : \mathbf{p}(\mathbf{p}(x,y),y) \in X\}.\end{aligned}$$

Φ' corresponds as operation on binary relations to Φ above, Φ'' is its inverse.

Let $\underline{\texttt{rn}}$ be defined as for **HAS**-formulas as in 7.2 relative to an assignment $X \mapsto X^*$ for second-order variables; and let $\underline{\texttt{rn}}'$ be the realizability notion as defined in this section, relative to an assignment $X \mapsto X^\circ$ of normal binary relations to the second-order variables (i.e. $x\,\underline{\texttt{rn}}'\,Xt := x \in [\![Xt]\!] := X^\circ(x,t)$; $x\,\underline{\texttt{rn}}'\,A := x \in [\![A]\!]$). Then for all formulas A of **HAS** there are ϕ_A, ψ_A such that

$$\begin{aligned}x\,\underline{\texttt{rn}}\,A(X_1,\ldots,X_n) &\to \psi_A\,\underline{\texttt{rn}}'\,A(X_1,\ldots,X_n)[X_1^\circ,\ldots,X_n^\circ/\Phi'X_1^*,\ldots,\Phi'X_n^*],\\ x\,\underline{\texttt{rn}}\,A(X_1,\ldots,X_n) &\to \phi_A\,\underline{\texttt{rn}}\,A(X_1,\ldots,X_n)[X_1^*,\ldots,X_n^*/\Phi''X_1^\circ,\ldots,\Phi''X_n^\circ],\end{aligned}$$

where $X_1,\ldots,X_n$ is a complete list of the second-order variables free in A. □

8.26. Lemma. *For Ω-sets* $\mathcal{X} \equiv (X,\sim)$, $\mathcal{Y} \equiv (Y,\sim')$, $\mathcal{Y}$ *separated, the elements of the Ω-functionset* $\mathcal{X} \to \mathcal{Y}$ *may be represented by functions* $f : X \to Y$.

Proof. Let $F : \mathcal{X} \times \mathcal{Y} \to \Omega$ be an arbitrary element of the Ω-exponent, for which EF is inhabited. Then for certain k, k'

$$\begin{aligned}&k \in \textstyle\bigcap_{x\in X}(Ex \to \bigcup_{y\in Y} Fxy),\\ &k' \in \textstyle\bigcap_{x\in X, y,y'\in Y}(Fxy \wedge Fxy' \to y \sim' y').\end{aligned}$$

By the second statement, it readily follows in combination with separatedness, that

$$Fxy \text{ inhabited, } Fxy' \text{ inhabited} \Rightarrow y = y',$$

and from the first statement that for all x with Ex inhabited, there is a y such that Fxy is inhabited; so let f be the function defined for x with Ex inhabited, such that $F(x, fx)$ is inhabited. □

8.27. Lemma. *For separated [effective]* $\mathcal{Y} \equiv (Y,\sim')$ *the Ω-function set* $(X,\sim) \to (Y,\sim')$ *is separated [effective].*

Proof. By the preceding lemma we may represent (there is a little checking to do, but we leave this to the reader) the elements of the exponent by the isomorphic $(X{\to}Y, \approx)$ with

$$\begin{aligned}&(f \approx g) \text{ inhabited} \Rightarrow f = g,\\ &f \approx f := \{m : \forall y \in Y \forall n \in Ey(m\bullet n \in E(fy))\},\end{aligned}$$

or combined into a single definition:

$$f \approx g := \{m : f = g \wedge \forall y \in Y \forall n \in Ey(m \bullet n \in E(fy))\}.$$

The separatedness has been built into the definition; as to proto-effectiveness, suppose $Ef \cap Eg$ inhabited, then for some m

$$\forall y \in Y \forall n \in Ey(m \bullet n \in E(fy) \cap E(gy));$$

by the proto-effectivity of $\mathcal{Y}$ it follows that $\forall y \in Y(fy = gy)$, i.e. $f = g$. □

REMARK. The fact that $\mathcal{X} \to \mathcal{Y}$ is separated for separated $\mathcal{Y}$ is also easily seen to hold for logical reasons (cf. 8.20): $\neg\neg f = g \leftrightarrow \neg\neg\forall x(fx = gx) \to \forall x(fx = gx) \leftrightarrow f = g$.

8.28. PROPOSITION. *The structure of functional ω-sets generated from $\mathcal{N}$ is isomorphic to* HEO *as defined in 3.3.*
Proof. Induction on the type structure. □
The following is immediate:

8.29. PROPOSITION. *For all function types σ generated from type 0 in* **HAH**, *the realizability interpretation validates a uniformity principle:*

$$\forall X^{\mathrm{P}[0]} \exists x^\sigma A(X, x) \to \exists x^\sigma \forall X^{\mathrm{P}[0]} A(X, x).$$

8.30. *Generalization to other kinds of realizability*

In the preceding section we have already indicated how the generalizations of realizabilities to second-order logic follow a pattern. If we combine this with the "truth-value semantics" idea introduced in the preceding section and used extensively above, we are led to consider other choices for Ω and Ω^*.

EXAMPLES. (a) If we want to generalize rnt-realizability, we take

$$\begin{aligned} \Omega^{\mathrm{rnt}} &:= \{(X, p) : X \subset \mathbb{N}, p \subset \{0\}\}, \\ \Omega^{\mathrm{rnt}*} &:= \{(X, p) \in \Omega : X \text{ inhabited, } 0 \in p\}. \end{aligned}$$

The crucial operations we have to define are $\wedge^{\mathrm{rnt}}$, $\to^{\mathrm{rnt}}$, $\bigcap^{\mathrm{rnt}}$:

$$\begin{aligned} (X, p) \wedge^{\mathrm{rnt}} (Y, q) &:= ((X \wedge Y, \{0 : 0 \in p \wedge 0 \in q\}), \\ (X, p) \to^{\mathrm{rnt}} (Y, q) &:= ((X \to Y), \{0 : 0 \in p \to 0 \in q\}), \\ \textstyle\bigcap^{\mathrm{rnt}}_{y \in Y}(Z_y, p_y) &:= (\textstyle\bigcap_{y \in Y} Z_y, \bigcap_{y \in Y} p_y). \end{aligned}$$

N.B. We do not really need to define an operation $\wedge^{\mathrm{rnt}}$, since in a second-order context we can define (7.3) operation $\wedge'$ in terms of the other operations, producing objects $(X, p) \wedge' (Y, q)$ isomorphic to $(X, p) \wedge^{\mathrm{rnt}} (Y, q)$.

(b) For modified realizability we can put

$$\Omega^{\mathrm{mrn}} := \{(X, Y) : X \subset Y \subset \mathbb{N}, 0 \in Y\},$$
$$\Omega^{\mathrm{mrn}*} := \{(X, Y) \in \Omega : X \text{ inhabited}\}.$$

$\bigcap^{\mathrm{mrn}}$ is defined component-wise, and for $\to^{\mathrm{mrn}}$ we take

$$(X, Y) \to^{\mathrm{mrn}} (X', Y') := ((X \to Y) \cap (X' \to Y'), X' \to Y')).$$

(In order to guarantee that 0 always occurs in the second component we must choose our gödelnumbering of the partial recursive functions such that $\Lambda x.0 = 0$.)

(c) For LifschitzLifschitz, V. realizability another idea is needed, a reformulation of the original definition which makes LifschitzLifschitz, V. realizability fit the general pattern (**?**).

8.31. *Notes*

The proper definition of realizability for higher-order logic emerged from the study of special toposes ((**?**), (**?**), (**?**), Grayson(**?**, **?**, **?**)). Aczel(**?**) described a less far-reaching common generalization of HeytingHeyting, A.-valued and realizability semantics. The higher-order extension of $\underline{\mathtt{rln}}$-realizability is due to (**?**). By means of this extension he shows that the following principle RP (*Richman'sRichman, F. Principle*)

$$\forall X^d (\forall Y^d (X \subset Y \vee X \cap Y = \emptyset \to \exists n \forall x (x \in X \to x = n))$$

(where $\forall X^d$, $\exists Y^d$ are quantifiers ranging over *decidable* subsets of $\mathbb{N}$) is false in the "LifschitzLifschitz, V. topos" and true in Eff, that is to say false in the higher-order extension of LifschitzLifschitz, V. realizability and true in the realizability interpretation for higher-order logic described in the preceding section. More information on the LifschitzLifschitz, V. topos Lif is given in (**?**).

9 Further work

9.1. *Realizability for set-theory*

It is also possible to define $\underline{\mathtt{rn}}$-realizability, or the abstract version $\underline{\mathtt{r}}$-realizability for the language of set theory. The definition is straightforward except for the fact that we have to build in extensionality.

The problem becomes clear if we try to extend the definition of $\underline{\mathtt{rn}}$-realizability given in 7.2 to intuitionistic third-order arithmetic $\mathbf{HAS}^3$ (variables $X^2, Y^2, \ldots$) in which we can also quantify over $\mathcal{P}\mathcal{P}(\mathbb{N})$, and with full impredicative comprehension and extensionality

EXT $\quad \forall X^2 (Y^1 \in X^2 \wedge Y^1 = Z^1 \to Z^1 \in X^2)$

where $X^1 = Y^1 := \forall z(z \in X \leftrightarrow z \in Y)$. If we take as clauses $x\,\underline{\mathtt{rn}}\,X^2(Y^1) := X^{*2}(Y^{*1}, x)$, $x\,\underline{\mathtt{rn}}\,\forall X^2.A(X^2) := \forall X^{*2}(x\,\underline{\mathtt{rn}}\,A(X^2)$, etc., we discover that there is no problem in proving soundness except for the axiom EXT; this imposes a restriction on the sets over which the "starred variables" X^{*2} should range.

Some authors, e.g. (**?**), solve the problem in the case of set theory by first giving a realizability interpretation for a set theory without the extensionality axiom, combined with an interpretation of the theory with extensionality into set theory without extensionality. Others such as (**?**) build the extensionality into the definition of realizability.

The earliest paper defining realizability for set theory is (**?**). Other papers using realizability for set theory are: (**?**), Friedman and Scedrov(**?**, **?**), McCarty(**?**, **?**), (**?**), and the series of papers by KhakhanyanKhakhanyan, V. Kh..

9.2. *Comparison with functional interpretations*

Another type of interpretation which is in certain respects analogous to (modified) realizability, but in other respects quite different, is the so-called Dialectica interpretation devised by (**?**). There is also a modification due to (**?**). As we have seen, modified realizability associates to formulas A of $\mathbf{HA}^\omega$ $\exists$-free formulas of the form $A_{\mathrm{mr}}(x^\sigma)$ (x^σ a new variable not free in A), expressing "x^σ modified-realizes A". The Dialectica- and DillerDiller, J.–NahmNahm, W. interpretation on the other hand associate with A formulas $\forall y^\tau A_D(x^\sigma, y^\tau)$ and $\forall y^\tau A_{DN}(x^\sigma, y^\tau)$ respectively, σ, τ depending on the logical structure of A alone, A_D, A_{DN} quantifier-free; we may read $\forall y^\tau A_D(x^\sigma, y^\tau)$, $\forall y^\tau A_{DN}(x^\sigma, y^\tau)$ as "x^σ D-interprets A"and "x^σ DN-interprets A" respectively.

For a soundness proof for the Dialectica interpretation, the prime formulas of the theory considered have to be *decidable* with a decision function of the appropriate type; for the DillerDiller, J.-NahmNahm, W. interpretation this is not necessary. For theories with decidable prime formulas (e.g. $\mathbf{I}$-$\mathbf{HA}^\omega$) the DillerDiller, J.-NahmNahm, W. interpretation is equivalent to the Dialectica interpretation. For background information the reader may consult the commentary to (**?**) in (**?**), and the relevant chapter elsewhere in this volume.

SteinStein, M. has constructed a whole sequence of interpretations intermediate between the DN-interpretation and modified realizability; see the papers by SteinStein, M., and (**?**).

9.3. *Formulas-as-types realizability*

In the formulas-as-types paradigm, formulas (representing propositions) are regarded as determined by (identified with) the set of their proofs. The idea is illustrated by taking a natural deduction formulation of intuitionistic predicate logic, and writing the deductions as terms in a typed lambda-calculus.

Normalization of the deductions suggests equations between the terms of such a calculus (in particular beta-conversion) and "t proves A" for compound A then behaves like an abstract realizability notion. Of particular interest is the realizability obtained by stripping the proof-terms of their types. With combinators instead of lambda-abstraction, such a realizability is already used in Staples(**?**, **?**). (**?**) uses this concept for an elegant version of Girard'sGirard, J.-Y. proof of the normalization theorem for second-order intuitionistic logic. For another version of the proof see (**?**).

(**?**, **?**) study completeness questions for such realizabilities. More specifically, one is interested in completeness results of the following type: for all formulas A of a certain formal system **S**, $\vdash_{\mathbf{S}} A$ iff $\vdash_{\mathbf{T}} \exists x(x\mathbf{r}A)$ where **T** is a suitable formal system (intuitionistic or classical logic), and **r** the abstract version of realizability studied.

"Formulas-as-types" has also been a leading idea in the formulation of various typed theories, such as the theories of Martin-Löf(**?**, **?**), permitting to absorb logical operations into type-forming operations (implication is subsumed under function-type formation, universal quantification under formation products of dependent types, etc.). In the proof-theoretic investigations of Martin-Löf'sMartin-Löf, P. type theories by de Swaen(**?**, **?**, **?**) realizability plays an important role.

9.4. *Completeness questions for realizabilities*

(**?**) gave an example of a classically valid, but not intuitionistically provable formula of propositional logic, such that all its arithmetical substitution examples are (classically) realizable; this result was improved by (**?**), who showed that the example also worked for <u>`rf`</u>- and <u>`mrf`</u>-realizability (in the latter case the substitution instances were provably realizable even intuitionistically). See also (**?**). On the other hand, (**?**) proved that the principle "Every formula for which all arithmetical substitution instances are realizable, is provable in intuitionistic propositional logic" is consistent with **HAS** (but not with **HAS** + M).

KleeneKleene, S.C. also showed that the class of formulas of *predicate* logic which are realizable under substitution is not recursive (for <u>`rn`</u>, <u>`rf`</u>, <u>`mrn`</u>). Similar questions have been studied at length in a series of papers by Plisko-Plisko, V.E.. A typical result of this kind is the following. Let $\mathcal{R}$ [$\mathcal{AR}$] be the class of all formulas $A(P_1, \ldots, P_n)$ of predicate logic such that all arithmetical substitution instances (i.e. formulas $A(P_1^*, \ldots, P_n^*)$ with $P_1^*, \ldots, P_n^*$ arithmetical) are <u>`rn`</u>-realizable [such that $\forall X_1 \ldots X_n\ A(X_1, \ldots, X_n)$ is <u>`rn`</u>-realizable as defined for **HAS**].

(**?**) showed that $\mathcal{AR}$ is a complete Π^1_1-set, and that $\mathcal{AR} \subset \mathcal{R}$; $\mathcal{R}$ is also not arithmetical as shown in (**?**).

Van Oosten(**?**), adapting a method originally due to de JonghJongh, D.H.J. de, gave a semantical proof of a result earlier established by proof-theoretic means by D. LeivantLeivant, D.: if all arithmetical substitution instances of a formula of predicate logic are provable in **HA**, the formula itself is a theorem of intuitionistic predicate logic("maximality of intuitionistic arithmetic"). The method uses a realizability in which rn-realizability and Beth-semantics are combined. His proof also yields the following completeness result for realizability. Let $\mathbf{HA}^+$ be an extension of **HA** obtained by adding to the language primitive constants $\bullet$ (application), **k**, **s** (combinators), with axioms saying that $(\mathbb{N}, \bullet, \mathbf{k}, \mathbf{s})$ is a partial combinatory algebra. Define r-realizability for $\mathbf{HA}^+$ relative to this combinatory algebra. Then a predicate formula A is provable in intuitionistic predicate logic iff all arithmetical instances of A are provably realizable in $\mathbf{HA}^+$.

A different sort of completeness result has been obtained by (**?**). He defined a modified realizability for predicate logic with a set-theoretic hierarchy as models for the finite-type functionals. All formula of predicate logic is realizable by an element of this hierarchy iff it is classically provable; but if we require that the realizing functionals are invariant under permutations of the basic domains, we obtain precisely the intuitionistically provable formulas. Inspection shows that the "modified" aspect of Läuchli'sLäuchli, H. construction is not really relevant. A modern recasting of Läuchli'sLäuchli, H. result, linking it with the category-theoretic interpretation of logic, was given by (**?**). See also (**?**).

9.5. *Realizability for subsystems of intuitionistic arithmetic*

(**?**) considers realizability for primitive recursive arithmetic, using primitive recursive functions instead of partial recursive functions. In particular the clauses for $\to$ and $\forall$ require modification. This is achieved using levels, which are provided by the Grzegorczyk hierarchy for the primitive recursive functions. In (**?**) the same idea is applied to obtain a realizability for **HA**, using so-called $< \varepsilon_0$–recursive functions instead of general recursive functions. (The $< \varepsilon_0$–recursive functions are precisely the functions provably recursive in **HA** and **PA**.) This permits reproving a number of metamathematical results for **HA** by realizability methods. The technique also applies to elementary analysis and finite–type arithmetic.

(**?**) uses rn- and rnt-realizability in a study of intuitionistic arithmetic with Σ_1-induction, $\mathrm{I}\Sigma_1$. Wehmeier shows that whenever $\vdash \forall n \exists m A(n,m)$ then there is a primitive recursive $t(x)$ such that $\forall n A(n, t(n))$. (**?**) use a version of rnt-realizability for a language of arithmetic extended with infinite disjunctions in a study of the intuitionistic counterpart IS^1_2 of Buss's system S^1_2 ((**?**)), and of $\mathrm{I}\Sigma_1$ just mentioned. The authors obtain a new proof of the fact, established in (**?**) by mrt-realizability, that whenever $\vdash \forall n \exists m A(n,m)$

in IS_2^1, then $\vdash \forall n A(n, t(n))$ where $t(x)$ is polynomial-time computable in x. They also sketch another proof of Wehmeier's result.

9.6. *Combining realizability with classical logic*

Lifschitz(**?**, **?**) considered an extension of classical arithmetic with an additional predicate $K(x)$, "x is computable". The result is a combination of classical arithmetic and realizability. It is to be noted that in the category Eff we can obtain something similar by considering side by side $\mathcal{N}$ and $\Delta\mathbb{N}$.

9.7. *Medvedev'sMedvedev, Yu. T. calculus of finite problems*

The calculus of finite problems as formulated by MedvedevMedvedev, Yu. T., is somewhat reminiscent of, but actually diverges rather far from recursive realizability. See the papers by MedvedevMedvedev, Yu. T., and by (**?**).

9.8. *Applications to Computer Science*

For some examples, see (**?**), (**?**) (realizability modeling of the theory of constructions), (**?**) (slash relations for type theory) and the papers by TatsutaTatsuta, M. (program synthesis by "realizability-cum-truth"). Within the effective topos one can find models for strong polymorphic type theories; see e.g. (**?**).

In (**?**) program-extraction from classical proofs of statements $\forall n \exists m A(n, m)$, A quantifier-free, is studied. Two methods are compared; one method is based on normalization of classical proofs formalized in a calculus of natural deduction, the other method uses modified realizability. The two methods yield terms $t_1(x), t_2(x)$ respectively such that $\forall n A(n, t_i(n))$ $(i = 1, 2)$ and for all numerals $\bar{n}$, $t_1(\bar{n}) = t_2(\bar{n})$. An ingredient in the second method is the combination of the *Gödel–Gentzen* translation ((**?**, 2.3)) with the Friedman-Dragalin *A-translation* ((**?**, 3.5)); by this combination a classical proof of $\forall n \exists m\, A(n, m)$, A quantifier-free, can be transformed into an intuitionistic proof of the same statement.

Interesting applications of the second method are given in (**?**). One of the examples concerns the following special case of Higman's lemma:

> If x^1, y^1 are two number-theoretic functions, we can find $n < m$ such that $x^1 n \leq x^1 m$ and $y^1 n \leq y^1 m$.

This fact is easily proved classically. Applying the program extraction via modified realizability, one obtains an algorithm which is in suitable cases quadratically faster than "brute–force" search. The syntactic definition of modified realizability and of the Friedman-Dragalin translation are used in a variant which helps keeping the complexity of the extracted program (term) down.

Abbreviations in the references

AML = *Annals of Mathematical Logic.*
AMS Transl. = *American Mathematical Society Translations, Series 2.*
APAL = *Annals of Pure and Applied Logic.*
Archiv = *Archiv für mathematische Logik und Grundlagenforschung.*
Doklady = *Doklady Akademii Nauk SSSR.*
Izv.Akad.Nauk = *Izvestiya Akademii Nauk SSSR. Seriya Matematicheskaya.*
JSL = *The Journal of Symbolic Logic.*
LMPS = *Logic, Methodology and Philosophy of Science.*
Math. Izv. = *Mathematics of the USSR, Izvestiya.*
SM = *Soviet Mathematics. Doklady.*
ZLGM = *Zeitschrift für Logik und Grundlagen der Mathematik*
Zapiski = *Zapiski Nauchnykh Seminarov Leningradskogo Otdeleniya Matematicheskogo Instituta imeni V. A. Steklova Akademii Nauk SSSR (LOMI).*