Constructive interpretations of logical and logical-mathematical languages

Valerii Plisko
Lomonosov Moscow State University, Russia
SCAN 2023
June 13, 2023

This is a survey of the author's main results obtained during his lifetime.

Kleene's recursive realizability

By constructive semantics we mean intuitionistic interpretations in which the concept of an effective operation is explained by means of the concept of an algorithm. The first semantics of this kind called *recursive realizability* was proposed by Kleene for the first-order language of arithmetic LA. The concept of recursive realizability originates in the informal *intuitionistic semantics* of mathematical sentences. From the intutionistic point of view, a sentence is *true* if it is *proved*. Thus the truth of a sentence is connected with its verification. For any true sentence A, we can consider its verification as a text justifying A.

This meaning of sentences leads to an original interpretation of logical connectives and quantifiers:

- a verification of $A \,\&\, B$ is a text containing a verification of A and a verification of B;
- a verification of $A \vee B$ is a text containing a verification of A or a verification of B and indicating which of them is justified;
- a verification of $A \to B$ is a text describing *a general effective operation* for obtaining a verification of B from each verification of A;
- a verification of $\neg A$ is a verification of the sentence $A \to \bot$, where $\bot$ is *a certainly absurd sentence* having no verification;
- a verification of $\forall x\, A(x)$ is a text describing a general effective operation for obtaining a verification of $A(a)$ for any a in the domain of the variable x;
- a verification of $\exists x\, A(x)$ is a text indicating an element a in the domain of x and containing a verification of $A(a)$.

In Kleene's recursive realizability the role of verifications are played by the natural numbers. A verification of an arithmetic sentence A is called *a realization* of A. Suppose that a Gödel enumeration of the unary partial recursive functions is fixed and $\{e\}$ denotes the function whose index is e. Then the relation "*a natural number e realizes an arithmetic sentence A*" is denoted $e \,\mathbf{r}\, A$ and is defined inductively:
if A is an atomic sentence, then $e \,\mathbf{r}\, A \rightleftharpoons [e = 0$ and A is true];
$e \,\mathbf{r}\, (A \,\&\, B) \rightleftharpoons [e = 2^a \cdot 3^b$, where $a \,\mathbf{r}\, A$, $b \,\mathbf{r}\, B]$;
$e \,\mathbf{r}\, (A \vee B) \rightleftharpoons [e = 2^0 \cdot 3^a$, where $a \,\mathbf{r}\, A$, or $e = 2^1 \cdot 3^b$, where $b \,\mathbf{r}\, B]$;
$e \,\mathbf{r}\, (A \to B) \rightleftharpoons \forall a\, [a \,\mathbf{r}\, A \Rightarrow \{e\}(a) \,\mathbf{r}\, B]$;
$e \,\mathbf{r}\, \neg A \rightleftharpoons [e \,\mathbf{r}\, (A \to 0 = 1)]$;
$e \,\mathbf{r}\, \forall x\, A(x) \rightleftharpoons \forall n\, \{e\}(n) \,\mathbf{r}\, A(n)$;
$e \,\mathbf{r}\, \exists x\, A(x) \rightleftharpoons [e = 2^n \cdot 3^a$, where $a \,\mathbf{r}\, A(n)]$.

An arithmetic sentence Φ is called *realizable* if $\exists e\, e \,\mathbf{r}\, \Phi$.

An arithmetic formula $\Phi(x_1, \ldots, x_m)$ with parameters $x_1, \ldots, x_m$ is called realizable iff there exists an m-ary recursive function f such that $f(k_1, \ldots, k_m) \, \mathbf{r} \, \Phi(k_1, \ldots, k_m)$ for every $k_1, \ldots, k_m$. In this case, we say that the function f realizes the formula $\Phi(x_1, \ldots, x_m)$.

Note that the formula $\Phi(x_1, \ldots, x_m)$ is realizable iff its universal closure $\forall x_1 \ldots \forall x_n \, \Phi(x_1, \ldots, x_m)$ is realizable.

For some classes of arithmetic sentences, recursive realizability is the same as the classical truth. Σ_1-*formula* is an arithmetic formula of the form $\exists \vec{x}\, A(\vec{x})$, where $A(\vec{x})$ is *a quantifier-free formula*. An arithmetic formula is called *almost negative* if it does not contain $\vee$ and contains $\exists$ only in Σ_1-subformulas. Any almost negative sentence is realizable iff it is classically true. In general, classical truth is not the same as realizability, but the set of realizable arithmetic sentences is recursively isomorphic to the set of classically true arithmetic sentences.

The notions of realizability for predicate formulas

On the basis of a constructive semantics for arithmetic formulas, one can define a corresponding semantics for predicate formulas.

The language of predicate logic contains a denumerable set of *the predicate variables* and a denumerable set of *the individual variables*.

Every predicate variable has definite *arity*.
Predicate variables of arity n are called n-*ary*.
0-ary predicate variables are called *propositional variables*.

Predicate formulas are constructed from propositional variables and *atomic formulas* of the form $P(x_1, \ldots, x_n)$, where P is an arbitrary n-ary variable ($n = 1, 2, \ldots$) and $x_1, \ldots, x_n$ are arbitrary individual variables, by means of *the logical connectives* &, $\vee$, $\rightarrow$, $\neg$, *the quantifiers* $\forall x$ and $\exists x$, where x is an arbitrary individual variable. $\mathfrak{A}(P_1, \ldots, P_n)$ will denote a predicate formula containing no predicate variables other than $P_1, \ldots, P_n$.

Propositional formulas are predicate formulas constructed from propositional variables using propositional connectives.

It would be natural to define *a realizable predicate formula* as a closed predicate formula $\mathfrak{A}(P_1, \ldots, P_n)$ such that every one of its arithmetic instances is realizable. But this definition admits several variants.

First, we can distinguish between *arbitrary* arithmetic instances of a closed predicate formula and only *closed* ones. Second, we can distinguish between *constructive* and *nonconstructive* treatment of the definition depending on the existence of an algorithm giving a realization for any arithmetic instance. Thus a priori we have four variants of realizability for predicate formulas. Two nonconstructive variants are following.

1) A closed predicate formula is called *weakly realizable* if every one of its closed arithmetic instances is realizable. Obviously, a propositional formula is weakly realizable iff it is a classical *tautology*. In particular, *the principle of excluded middle* $P \vee \neg P$ is weakly realizable. But in general, the weakly realizable predicate formulas are not the same as the classically valid formulas. For example, the classically valid predicate formula $\forall x\,(P(x) \vee \neg P(x))$ is not weakly realizable.

Let $Rw(x)$ mean that x is the Gödel number of a weakly realizable predicate formula.

2) A closed predicate formula is called *irrefutable* if every one of its arithmetic instances is realizable. Let $\mathfrak{A}(P_1, \ldots, P_n)$ be a predicate formula, $\Phi_1, \ldots, \Phi_n$ be a list of arithmetic formulas such that the formula $\mathfrak{A}(\Phi_1, \ldots, \Phi_n)$ is not realizable. In this case, we say that the list $\Phi_1, \ldots, \Phi_n$ is *a refutation* of this formula. Thus a predicate formula is irrefutable iff it has no refutation. The formula $P \vee \neg P$ is refutable, namely its refutation consisits of a Σ_1-formula $\mathsf{S}_0(x)$ expressing the predicate $\{x\}(x) = 0$, because the realizability of the formula $\forall x(\mathsf{S}_0(x) \vee \neg \mathsf{S}_0(x))$ would mean decidability of the predicate $\{x\}(x) = 0$. Obviously, any irrefutable predicate formula is weakly realizable.

Let $Ri(x)$ mean that x is the Gödel number of an irrefutable predicate formula.

Now consider constructive variants of the notion of realizable predicate formula.

3) A closed predicate formula is called *effectively realizable* if there exists an algorithm producing a realization for every closed arithmetic instance of this formula. Obviously, any effectively realizable predicate formula is irrefutable.
Let $Re(x)$ mean that x is the Gödel number of an effectively realizable predicate formula.

4) The fourth notion of realizability for predicate formulas means that for any (not necessary closed) arithmetic instance of a predicate formula one can effectively find a Gödel number of a general recursive function realizing this instance. It is rather obvious that this notion coincides with the notion of effectively realizable predicate formula.

The next variant of the notion of a realizable predicate formula is implied by Nelson's proof of the theorem that intuitionistic arithmetic HA is sound with respect to realizability. Namely, if a closed predicate formula is deducible in intuitionistic predicate calculus IQC, then there exists a natural number realizing any closed arithmetic instance of this formula.

5) A closed predicate formula is called *uniformly realizable* if there exists a natural number realizing every closed arithmetic instance of this formula. Evidently, every uniformly realizable predicate formula is effectively realizable.

Let $Ru(x)$ mean that x is the Gödel number of a uniformly realizable predicate formula.

It was noted above that every predicate formula deducible in IQC is uniformly realizable.
Let D be the propositional formula $\neg P \vee \neg Q$. Rose proved that the formula $((\neg\neg D \to D) \to (\neg D \vee \neg\neg D)) \to (\neg D \vee \neg\neg D)$ is uniformly realizable but is not deducible in the intuitionistic propositional calculus IPC.

Thus we have four notions of realizability for predicate formulas.
A closed predicate formula $\mathfrak{A}$ is
- weakly realizable if every one of its closed arithmetic instances is realizable;
- irrefutable if any its arithmetic instance is realizable;
- effectively realizable if there exists an algorithm allowing to find a realization of any closed arithmetic instance of $\mathfrak{A}$;
- uniformly realizable if there exists a natural number realizing any closed arithmetic instance of $\mathfrak{A}$.

The following theorem was proved by the author.

Theorem 1. *1) There is an effectively realizable predicate formula which is not uniformly realizable.*

2) There is an irrefutable predicate formula which is not effectively realizable.

3) There is a weakly realizable predicate formula which is not classically valid.

The problem of relations between irrefutable, effectively realizable, and uniformly realizable poropositional formulas is still open. Every one of the known realizable propositional formulas is in fact uniformly realizable and every one of the known non-realizable propositional formulas has a refutation.

Realizability and Medvedev logic

In 1932, Kolmogorov proposed an interpretation of intuitionistic logic as the logic of problems. 30 years later, this idea was somewhat refined by Medvedev in the form of the concept of finite validity for propositional formulas.

In 1963, Medvedev published a proof of the claim that every realizable propositional formula is finitely valid. The author has constructed a counterexample to this statement.

Theorem 2. *There is a uniformly realizable propositional formula which is not finitely valid.*

In the proof, the finite model property of the Medvedev logic and Yankov's characteristic formulas were significantly used.

The results on partial completeness of IPC

Rose proved that IPC is complete relative to realizability for formulas without $\to$. Namely, if such a formula is not deducible, then it has a refutation. Medvedev proved the completeness of the fragments of IPC without negation or disjunction relative to finite validity. From the above-mentioned erroneous theorem, a similar result was obtained for the propositional logic of recursive realizability.

The author proved that partial completeness really holds.

Theorem 3. *If a propositional formula A does not contain $\neg$ or does not contain $\vee$ and* IPC $\nvdash A$*, then A has a refutation.*

The nonarithmeticity of predicate realizability logic

Kleene proved that the set of irrefutable predicate formulas is not decidable. A further characterization of predicate realizability logic is given by the following theorem.

Theorem 4. *The predicates* $Rw(x)$, $Ri(x)$, $Re(x)$, *and* $Ru(x)$ *are not arithmetical, i.e., are not definable in the language* LA.

Theorem 4 is an immediate consequence of the nonarithmeticity of the set of the Gödel numbers of realizable arithmetic formulas and the following proposition.

Theorem 5. *For any arithmetic sentence F, one can effectively construct a closed predicate formula F^* such that*

1) if F is realizable, then F^* is uniformly realizable;

2) if F^* is weakly realizable, then F is realizable.

The proof is based on ideas related to the Tennenbaum theorem that there are no recursive non-standard models of arithmetic.

Theorem on schemes

Theorem 5 makes possible to imbed arithmetic with realizability semantics into predicate realizability logic. One can go farther and obtain the following construction technically comfortable for applications.

The notion of a scheme over the language of arithmetic was introduced by Kipnis. Schemes are formulas in the mixed language of arithmetic and predicate logic. A scheme can be considered as an arithmetic formula with predicate variables or as a predicate formula containing predicate constants. The notions of weak, effective, uniform realizabilities and irrefutability for the schemes are defined in the same way as in the case of predicate formulas.

The following *theorem on schemes* holds.

Theorem 6. *For any closed scheme $\mathfrak{A}$ it is possible to construct effectively a closed predicate formula $\mathfrak{A}^*$ such that*
1) $\mathfrak{A}$ is weakly realizable iff $\mathfrak{A}^*$ is weakly realizable;
2) $\mathfrak{A}$ is irrefutable iff $\mathfrak{A}^*$ is irrefutable;
3) $\mathfrak{A}$ is effectively realizable iff $\mathfrak{A}^*$ is effectively realizable;
4) $\mathfrak{A}$ is uniformly realizable iff $\mathfrak{A}^*$ is uniformly realizable.

This theorem made it possible to show the difference between the variants of realizability for predicate formulas: it is enough to find a scheme which is realizable in one sense but not realizable in another sense. We know that the notions of weakly realizable and irrefutable predicate formula are different. However there is a close relation between them.

Theorem 7. *The sets of weakly realizable predicate formulas and irrefutable predicate formulas are recursively isomorphic.*

A constructive predicate calculus

It follows from the Nelson theorem that the calculus IQC is sound with respect to recursive realizability. It was shown by Rose that the calculus IPC is not complete with respect to this semantics. Subsequently, Markov formulated a logical principle, now called the Markov principle, meaning, in particular, the realizability of the predicate formula

$$\forall x\,(P(x) \vee \neg P(x)) \to (\neg\neg\exists x\, P(x) \to \exists x\, P(x))$$

(denote it M), non-deducible in IQC.

A scheme ECT over the arithmetic language, called the extended Church thesis, is sound with respect to the semantics of recursive realizability. In the presence of Markov's principle, this scheme is equivalent to the scheme

$$\forall x\,(\neg A(x) \to \exists y\,B(x,y)) \to$$

$$\to \exists z\,\forall x\,(\neg A(x) \to \exists y\,(\{z\}(x) = y\,\&\,B(x,y))).$$

By the scheme theorem, ECT can be replaced with a predicate formula ECT^*. The calculus MQC $=$ IQC $+\ M + ECT^*$ can be considered as a possible constructive predicate calculus. Note that this calculus is not intermediate between intuitionistic and classical calculi. The arithmetic theory based on the calculus MQC and the Peano axioms proves the same theorems as the Markov arithmetic MA $=$ HA $+\ M + ECT$.

LA_1-realizability of predicate formulas

We have considered some variants of constructive semantics for predicate formulas based on the notion of recursive realizability for the first-order language of arithmetic. But in connection with the problem of studying constructive predicate logic, it is of interest to consider also richer languages with constructive semantics. As an example, one can consider the language LA_1 being an extension of the language of formal arithmetic LA with a unary predicate symbol T denoting the truth predicate for LA.

A constructive semantics of the language LA_1 is defined under the assumption that a constructive semantics of the language LA is known. It is sufficient to explain only the meaning of atomic formulas of the form $T(t)$, where t is a closed term. Let a number n be the value of the term t. Then the sentence $T(t)$ is considered as constructively true iff n is the Gödel number of a constructively true arithmetic sentence. Semantics of the language LA_1 can be described also in terms of realizability. Namely, $e \,\mathsf{r}\, T(t)$ iff the value of the term t is the Gödel number of an arithmetical sentence Φ and $e \,\mathsf{r}\, \Phi$. Realizability of other formulas is defined inductively in the same way as for the formulas of the language LA.

Obviously, the predicate $T(x)$ can not be defined in LA. Various notions of LA_1-realizability for predicate formulas and schemes are defined by replacing arithmetic instances with LA_1-instances in the definitions of a weakly realizable, irrefutable, effectively realizable, and uniformly realizable predicate formula.

Theorem 8. *Let $\mathfrak{A}$ be a closed scheme, $\mathfrak{A}^*$ be the predicate formula constructed in the proof of Theorem 6. Then the scheme $\mathfrak{A}$ is weakly LA_1-realizable (LA_1-irrefutable, effectively, uniformly LA_1-realizable) iff $\mathfrak{A}^*$ is weakly LA_1-realizable (respectively, LA_1-irrefutable, effectively, uniformly LA_1-realizable).*

The following analog of Theorem 5 also holds.

Theorem 9. *For any sentence F of the language LA_1 one can effectively construct a closed predicate formula F^* such that*
1) if F is realizable, then F^* is uniformly LA_1-realizable;
2) if F^* is weakly LA_1-realizable, then F is realizable.

It follows from Theorem 9 that the set of Gödel numbers of LA_1-realizable in any sense closed predicate formulas is not definable in the language LA_1. On the other hand, the sets of Gödel numbers of weakly realizable, irrefutable, effectively realizable, and uniformly realizable predicate formulas are definable in the language LA_1. This means that the notions of realizability of a predicate formula based on interpreting predicate variables by formulas of the languages LA and LA_1 are essentially different.

This fact can be established also in a direct way.

Theorem 10. *There exists a uniformly realizable closed predicate formula which is not weakly LA_1-realizable.*

The following proposition concerning a relation between usual realizability and LA_1-realizability is of interest.

Theorem 11. *Any LA_1-irrefutable predicate formula is uniformly realizable.*

Transfinite sequences of constructive predicate logics

We see that the notion of realizable predicate formula essentially depends on a language for formulating predicates substituted for predicate variables, therefore the semantics of predicate logic based on constructive semantics of the first-order language of arithmetic is rather occasional. For discovering general logical principles acceptable from the constructive point of view, we are forced to consider languages richer than the language of arithmetic. We considered the language LA_1 and found that the set of constructively valid predicate formulas is essentially narrowed down by considering a richer language. The procedure of extending languages by adding the truth predicate can be continued.

Let L be a first-order language obtained by adding a binary predicate symbol T to the language of arithmetic LA. Let us call L *the universal language*. Let $\prec$ be *a recursive well-ordering* of (an initial segment of) the set of natural numbers $\mathbb{N}$. The order-type of $\prec$ is *a constructive ordinal* α. Then every natural number m is a notation of an ordinal $\beta < \alpha$; this fact will be denoted by $|m| = \beta$. Without loss of generality one can set $|0| = 0$.

For any $\beta \leq \alpha$ a fragment LA_β of the universal language L is defined. Formulas of the language LA_β are constructed in the usual way from atomic formulas of the form $t_1 = t_2$, where t_1 and t_2 are terms, and $T(n, t)$, where $|n| < \beta$, t is a term. Note that LA_0 is just the language of arithmetic LA. If $\gamma < \beta$, then every LA_γ-formula is also an LA_β-formula and LA_α is the union of the languages LA_β for $\beta \leq \alpha$.

Constructive semantics of the languages LA_β is defined in terms of realizability by *transfinite induction* on β. Let us assume that for every $\gamma < \beta$ the relation "*a natural number* e *is a* γ*-realization of an* LA_γ*-sentence* Φ" denoted by $e\,\mathsf{r}_\gamma\,\Phi$ is defined. Then the relation $e\,\mathsf{r}_\beta\,\Phi$ is defined by induction on the number of logical connectives and quantifiers in Φ. If Φ is an atomic sentence of the form $T(m,t)$, where $|m| = \gamma < \beta$ and t is a term without variables whose value is a number n, then $e\,\mathsf{r}_\beta\,\Phi$ iff n is the Gödel number of an LA_γ-sentence Ψ and $e\,\mathsf{r}_\gamma\,\Psi$. Otherwise the definition of the relation $e\,\mathsf{r}_\beta\,\Phi$ does not differ from Kleene's definition of the relation $e\,\mathsf{r}\,\Phi$. It is easily shown that if Φ is an LA_γ-sentence and $\gamma < \beta$, then $e\,\mathsf{r}_\gamma\,\Phi$ iff $e\,\mathsf{r}_\beta\,\Phi$. Thus one can write simply $e\,\mathsf{r}\,\Phi$ instead of $e\,\mathsf{r}_\beta\,\Phi$. Therefore, if Φ is an LA_γ-sentence, then $e\,\mathsf{r}\,\Phi$ means that $e\,\mathsf{r}_\beta\,\Phi$ for any β such that $\gamma \leq \beta \leq \alpha$. In particular, $e\,\mathsf{r}\,\Phi$ means the same as $e\,\mathsf{r}_\alpha\,\Phi$.

Weak LA_β-realizable, LA_β-irrefutable, effectively LA_β-realizable, and uniformly LA_β-realizable predicate formulas are defined on the base of realizability for the language LA_β, where $\beta \le \alpha$, in a natural way. For these notions, the same results as for the usual realizability and LA_1-realizability can be proved. In particular, the following propositions hold.

Theorem 12. *The set of the Gödel numbers of LA_β-realizable (in any sense) predicate formulas is not definable in the language LA_β.*

Theorem 13. *If $\gamma < \beta < \alpha$, then there exists an LA_γ-uniformly realizable closed predicate formula which is not weakly LA_β-realizable.*

Absolute realizability

The fact that the concept of a realizable predicate formula depends on the language in which we formulate the predicates which are substituted for predicate variables leads to the problem of developping a concept of realizability for predicate formulas that includes interpretations relative to concrete languages with constructive semantics as special cases. We see that constructive predicate logic can not be identified with the theory of LA_α-realizability for any constructive ordinal α. Let us look at the problem from the classical point of view.

An important idea of constructive semantics of mathematical sentences is that the meaning of a sentence is connected with a construction that verifies its truth. In classical logic a sentence is identified whith its truth value. Similarly, we can identify the constructive meaning of a sentence with the set of the Gödel numbers of the objects verifying the truth of this sentence. So we come to the idea of interpreting sentences as in general arbitrary sets of natural numbers.

In this case, the logical operations are defined according to recursive realizability.

- $A \,\&\, B \leftrightharpoons \{2^a \cdot 3^b \mid a \in A,\, b \in B\}$;
- $A \vee B \leftrightharpoons \{2^0 \cdot 3^a \mid a \in A\} \cup \{2^1 \cdot 3^b \mid b \in B\}$;
- $A \to B \leftrightharpoons \{x \mid \forall a\, (a \in A \Rightarrow \{x\}(a) \in B)\}$;
- $\neg A \leftrightharpoons A \to \emptyset$.

The concept of a predicate is generalized in a corresponding way. Namely a k-ary *generalized predicate* is a function (in the set-theoretical sense) of type $\mathbb{N}^k \to 2^{\mathbb{N}}$, i. e., a function defined on $\mathbb{N}^k$ whose values are sentences (sets of natural numbers). Applying quantifiers to generalized predicates is defined in a natural way. Let P be an unary generalized predicate. Then

- $\forall x\, P(x) \leftrightharpoons \{x \mid \forall a\, \{x\}(a) \in P(a)\}$;

- $\exists x\, P(x) \leftrightharpoons \{2^a \cdot 3^b \mid b \in P(a)\}$.

Let L be a first-order extension of LA with realizability semantics. We say that an n-ary generalized predicate $\mathcal{P}$ is definable in L if there is an L-formula $A(x_1, \ldots, x_n)$ such that

$$\forall k_1, \ldots, k_n\, [\mathcal{P}(k_1, \ldots, k_n) = \{e \mid e\,\mathsf{r}\, A(k_1, \ldots, k_n)\}].$$

Let $\mathfrak{A}$ be a closed predicate formula with predicate variables $P_1, \ldots, P_n$. Let $\mathcal{P}_1, \ldots, \mathcal{P}_n$ be a system of generalized predicates, with $\mathcal{P}_i$ an m_i-ary predicate if P_i is an m_i-ary predicate variable (such systems will be called admissible for $\mathfrak{A}$). Regarding the predicate variables $P_1, \ldots, P_n$ as notations for the generalized predicates $\mathcal{P}_1, \ldots, \mathcal{P}_n$ and implementing operations over them prescribed by the formula $\mathfrak{A}$, we obtain a set of natural numbers $\mathfrak{A}(\mathcal{P}_1, \ldots, \mathcal{P}_n)$ as a result of interpreting the formula $\mathfrak{A}$ on the system of generalized predicates $\mathcal{P}_1, \ldots, \mathcal{P}_n$.

A closed predicate formula $\mathfrak{A}$ is *absolutely uniformly realizable* iff there exists a number e such that $e \in \mathfrak{A}(\mathcal{P}_1, \ldots, \mathcal{P}_n)$ for any generalized predicates $\mathcal{P}_1, \ldots, \mathcal{P}_n$ admissible for $\mathfrak{A}$.

Suppose that a closed predicate formula $\mathfrak{A}(P_1, \dots, P_n)$ does not contain an individual variable z. For any m_i-ary predicate variable P_i $(i = 1, \dots, n)$ let P_i' be an $(m_i + 1)$-ary predicate variable. Replace in the formula $\mathfrak{A}$ every atomic subformula of the form $P_i(v_1, \dots, v_k)$ by $P_i'(z, v_1, \dots, v_k)$. Denote the obtained formula by $\mathfrak{A}(z)$. It is obvious that a predicate formula $\mathfrak{A}$ is irrefutable iff every one of the closed instances of the formula $\forall z\, \mathfrak{A}(z)$ is realizable, i. e., the formula $\forall z\, \mathfrak{A}(z)$ is weakly realizable. This observation leads to the following definitions.

A system of generalized predicates $\mathcal{P}_1, \dots, \mathcal{P}_n$ admissible for the formula $\forall z\, \mathfrak{A}(z)$ is called *a refutation* of the formula $\mathfrak{A}$ iff

$$\forall z\, \mathfrak{A}(z)(\mathcal{P}_1, \dots, \mathcal{P}_n) = \emptyset.$$

A closed predicate formula $\mathfrak{A}$ is called *absolutely irrefutable* if no system of generalized predicates is a refutation of $\mathfrak{A}$.

Obviously, any absolutely uniformly realizable predicate formula is absolutely irrefutable. The converse also holds.

Theorem 14. *If a predicate formula is not absolutely uniformly realizable, then it has a refutation.*

A closed predicate formula is called *absolutely realizable* if it is absolutely uniformly realizable or, what is the same, absolutely irrefutable.

The notion of an effectively realizable predicate formula has no immediate analog in the framework of absolute realizability because generalized predicates are not constructive objects. But if we consider that the notion of an effectively realizable predicate formula occupies an intermediate position between the notions of a uniformly realizable and an irrefutable predicate formula and Theorem 14, we can say that every absolutely irrefutable formula is effectively absolutely realizable in the strongest sense: there exists a unique realization for all instances of this formula obtained by substituting generalized predicates for the predicate variables.

The class of absolutely realizable predicate formulas is closed under derivability in the constructive predicate calculus MQC introduced above. In particular, it contains all closed formulas deducible in this calculus. This class has the disjunction property: if a formula $\mathfrak{A} \vee \mathfrak{B}$ is absolutely realizable, then at least one of the formulas $\mathfrak{A}$ and $\mathfrak{B}$ is absolutely realizable. This directly follows from the fact that every absolutely realizable predicate formula is uniformly absolutely realizable. There are absolutely realizable predicate formulas which are not classically valid.

The class of absolutely realizable predicate formulas is nonarithmetical. This statement is proved by the same argument as the nonarithmeticity of the realizable predicate formulas. Further characterization of the predicate logic of absolute realizability is obtained by studying its relation to the second-order arithmetic. *The language of the second-order arithmetic* is an extension of LA by means of adding functional variables and quantifiers on them. In the intended semantics of this language, arbitrary number functions are considered as possible values of functional variables. Π^1_1-sentence is a closed formula of the second-order arithmetic of the form $\forall f_1 \ldots \forall f_n \mathfrak{A}$, where $f_1, \ldots, f_n$ are functional variables, $\mathfrak{A}$ is a formula without functional quantifiers.

Theorem 15. *The set of absolutely realizable predicate formulas is recursively isomorphic to the set of true Π^1_1-sentences of the second-order arithmetic.*

The set of true Π^1_1-sentences of the second-order arithmetic is Π^1_1-complete, i. e., it is a Π^1_1-set and every Π^1_1-set is 1-reducible to it. Combining this with Theorem 15, we get the following proposition.

Theorem 16. *The set of Gödel numbers of absolutely realizable predicate formulas is Π^1_1-complete.*

Π^1_1-complete sets are isomorphic one to another. Thus Theorem 16 gives a final characterization of the set of absolutely realizable predicate formulas from the point of view of recursion theory.

In the definition of an absolutely realizable predicate formula arbitrary generalized predicates are admissible for interpreting predicate variables. In total, there is a continuum of generalized predicates. On the other hand, the set of predicate formulas is denumerable, thus there exists a denumerable class of generalized predicates containing refutations for predicate formulas which are not absolutely realizable. The problem of searching such a class can be made more precise in the following way.

Let $\mathcal{K}$ be a class of generalized predicates. We say that a closed predicate formula $\mathfrak{A}(P_1, \ldots, P_n)$ is $\mathcal{K}$-*uniformly realizable* iff there exists a number e such that $e \in \mathfrak{A}(\mathcal{P}_1, \ldots, \mathcal{P}_n)$ for any system of generalized predicates $\mathcal{P}_1, \ldots, \mathcal{P}_n$ in the class $\mathcal{K}$ admissible for substitution in $\mathfrak{A}$. A closed predicate formula $\mathfrak{A}$ will be called $\mathcal{K}$-*irrefutable* if no system of generalized predicates in the class $\mathcal{K}$ is a refutation of the formula $\mathfrak{A}$. A class of generalized predicates $\mathcal{K}$ will be called *a basis for absolute realizability* iff every $\mathcal{K}$-irrefutable predicate formula is absolutely realizable.

Note that for any constructive ordinal α, the class of generalized predicates definable in the language LA_α can not be a basis for absolute realizability.

We say that a generalized predicate $\mathcal{P}$ is *hyperarithmetical* iff an $(m+1)$-ary relation $x_{m+1} \in \mathcal{P}(x_1, \ldots, x_m)$ is hyperarithmetical, i. e., is expressed both by a Σ^1_1-formula and a Π^1_1-formula of the second-order arithmetic.

Theorem 17. *The class $\mathcal{H}$ of all hyperarithmetical generalized predicates is not a basis for absolute realizability.*

Now consider a class of generalized predicates which is a basis for absolute realizability. Let $W(x)$ be a Π^1_1-complete predicate, $LA + W$ be an extension of LA by adding an unary predicate symbol W. The notion of recursive realizability is extended in the following way: a natural number e realizes an atomic formula $W(t)$, where t is a term without variables, iff $e = 0$ and $W(n)$ holds, where n is the value of the term t.

Theorem 18. *A closed predicate formula is absolutely realizable iff it has no $(LA + W)$-refutation.*

Modified realizability

In the informal explanation of the intuitionistic semantics, it is used the notion of "a general effective operation" that, from the mathematical point of view, can not be conceived otherwise than a mapping. Therefore it seems natural, in a precise specification of a variant of intuitionistic semantics of arithmetic sentences, to use a language making possible to speak about arbitrary mappings. Taking into account necessity of a treatment, for example, of sentences of the form $(A \to B) \to C$, one needs to consider along with mappings from $\mathbb{N}$ to $\mathbb{N}$, also mappings from the set of mappings of type $\mathbb{N} \to \mathbb{N}$ to $\mathbb{N}$ and not only them. This can be made by means of *the language of the arithmetic of finite types* LA^{ω}.

Types are expressions defined inductively in the following way:
1) 0 is a type;
2) if σ and τ are types, then $(\sigma \to \tau)$ is a type.
The language LA^ω is an extension of the arithmetic language LA obtained by adding an infinite set of variables of any type.
Terms of type σ are defined as follows:
1) variables of type σ are terms of type σ;
2) if s is a term of type $(\tau \to \sigma)$ and t is a term of type τ, then $s(t)$ is a term of type σ.
Atomic formulas of the language LA^ω are of the form $t_1 = t_2$, where t_1 and t_2 are terms of type 0. More complicated formulas are constructed from atomic ones in the usual way by means of logical connectives and quantifiers on variables of *any type*.

An interpretation M of the language LA^ω is defined by assigning to every type τ, a nonempty set M_τ *of the objects of type* τ considering as a domain of variables of type τ; in this case, $M_0 = \mathbb{N}$. For any pair of types σ, τ it is specified a mapping $\mathsf{app}_{\sigma,\tau} : M_{(\sigma\to\tau)} \times M_\sigma \to M_\tau$ defining the application of any object a of type $(\sigma \to \tau)$ to any object b of type σ resulting in an object $\mathsf{app}_{\sigma,\tau}(a, b)$ of type τ denoted also by $a(b)$. Thus with any object a of type $(0 \to 0)$, it is *associated* a function $f_a : \mathbb{N} \to \mathbb{N}$ such that $f_a(x) = \mathsf{app}_{0,0}(a, x)$ for all $x \in \mathbb{N}$. If an interpretation M of the language LA^ω is given, it is possible to speak about true and false closed formulas of this language in the interpretation M assuming the usual classical meaning of logical connectives and quantifiers.

An example of an interpretation of the language LA^ω is the set-theoretical interpretation M, where $M_0 = \mathbb{N}$ and $M_{(\sigma\to\tau)} = M_\tau^{M_\sigma}$ is the set of *all* mappings from M_σ to M_τ. There are constructive interpretations of this language, i. e., such that the objects of type $(\sigma \to \tau)$ define *computable* mappings from M_σ to M_τ.

One of such interpretations is *the system of hereditary recursive operations* HRO defined as follows. For any type σ the set HRO_σ consists of natural numbers. Of course, $HRO_0 = \mathbb{N}$. Further, if HRO_σ and HRO_τ are defined, then $HRO_{(\sigma\to\tau)}$ is the set of natural numbers x such that the function $\{x\}$ transforms any element $a \in HRO_\sigma$ into an element $\{x\}(a) \in HRO_\tau$.

Several constructive semantics can be obtained by means of a generalization of recursive realizability.

Modified realizability introduced by Kreisel is a translation from LA into LA^ω revealing intuitionistic meaning of sentences. For any LA-formula Φ it is defined inductively an LA^ω-formula $\mathbf{x}\,\mathsf{mr}\,\Phi$, where $\mathbf{x}$ is a (possibly empty) list of variables of the language LA^ω. For an atomic Φ the list $\mathbf{x}$ is empty and $\mathbf{x}\,\mathsf{mr}\,\Phi$ is Φ. Consider implication and universal quantifier. Assume that the list $\mathbf{x}$ in $\mathbf{x}\,\mathsf{mr}\,\Phi$ contains only one variable x of type σ and the list $\mathbf{y}$ in $\mathbf{y}\,\mathsf{mr}\,\Psi$ contains only one variable y of type τ. Then the translation of the formula $\Phi \to \Psi$ is $\forall x\,(x\,\mathsf{mr}\,\Phi \to u(x)\,\mathsf{mr}\,\Psi)$ and the corresponding list of variables consists of one variable u of type $(\sigma \to \tau)$. The translation of the formula $\forall v\,\Phi$, where v is a variable of type ρ, is $\forall v\, w(v)\,\mathsf{mr}\,\Phi$ and the corresponding list of variables consists of one variable w of type $(\rho \to \sigma)$.

Let $\mathsf{mr}\,\Phi$ denote the formula $\exists \mathbf{x}\,\mathbf{x}\,\mathsf{mr}\,\Phi$.

Let M be an interpretation of the language LA^ω.
An arithmetic sentence Φ will be called M-*realizable* iff the formula mr Φ is true in the interpretation M. The set of M-realizable arithmetic sentences is denoted by MR(M).

For any interpretation M, the theory MR(M) is complete.

The Gödel interpretation

An interprertation of arithmetic sentences proposed by Gödel reveals constructive point of view in another way. Gödel has defined a translation from the language LA into the language LA^ω that transforms every arithmetic formula Φ into a formula Φ^D of the language LA^ω having the form $\exists \mathbf{x} \forall \mathbf{y}\, \Phi_D(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ and $\mathbf{y}$ are (possibly empty) lists of variables of the language LA^ω and $\Phi_D(\mathbf{x}, \mathbf{y})$ is a quantifier-free formula of this language.

For an atomic Φ formulas Φ^D and Φ_D coincide with Φ and the lists $\mathbf{x}$ and $\mathbf{y}$ are empty.

Consider the case of constructing more complicated formulas by means of implication and universal quantifier.
For the sake of simplicity, we shall assume that Φ^D is of the form $\exists x \, \forall y \, \Phi_D(x, y)$, where x is a variable of type σ, y is a variable of type τ, and Ψ^D has the form $\exists u \, \forall v \, \Psi_D(u, v)$, where u is a variable of type ρ, v is a variable of type π. Then the formula $(\Phi \to \Psi)^D$ is of the form $\exists w \, \exists z \, \forall x \, \forall v \, (\Phi \to \Psi)_D$, where w is a variable of type $(\sigma \to \rho)$, z is a variable of type $(\sigma \to (\pi \to \tau))$, and $(\Phi \to \Psi)_D$ is $\Phi_D(x, z(x)(v)) \to \Psi_D(w(x), v)$.

The formula $(\forall u \, \Phi)^D$, where u is a variable of type ρ, has the form $\exists s \, \forall u \, \forall y \, (\forall u \, \Phi)_D$, where s is a variable of type $(\rho \to \sigma)$ and $(\forall u \, \Phi)_D$ is the formula $\Phi_D(s(u), y)$.

As in the case of modified realizability, we obtain the whole spectrum of semantics for arithmetic sentences.

For any interpretation M of the language LA^ω, an arithmetic sentence Φ will be called M-*true by Gödel* if the formula Φ^D is true in the interpretation M.

The set of M-true by Gödel sentences is denoted by D(M).

Note that the theory D(M) commonly is not complete.

Predicate logics of constructive theories

Once Mints asked the author what is the situation with the predicate logic based on the Gödel interpretation. Later it turned out that he meant only a set-theoretic interpretation, and in this case the answer is trivial: such logic coincides with the classical one. However, the author was interested in constructive semantics. Since Gödel's translation is much more complicated than recursive realizability, any intuitive approach does not work here at all. The author had to formalize in a rather general way the methods of investigating predicate logics based on the application of ideas related to the Tennenbaum theorem.

Let T be an arithmetic theory. A closed predicate formula A is said to be T-valid if every closed arithmetic instance of A is in T. Call the set of all T-valid formulas the predicate logic of the theory T and denote $\mathcal{L}(T)$. It is easy to prove that the predicate logic of Peno arithmetic PA is just the classical predicate logic. However it was shown by Yavorski that the predicate logics of some classical theories are wider than classical predicate logic. We shall be interested in predicate logics of constructive arithmetic theories.

A constructive arithmetic theory is an arbitrary set of arithmetic sentences closed under deducibility in the theory

$$\mathsf{CHA} = \mathsf{HA} + M + CT,$$

where CT is the scheme

$$\forall x\, \exists y\, A(x,y) \to \exists z\, \forall x\, \exists y\, (\{z\}(x) = y\, \&\, A(x,y)).$$

For example, the Markov arithmetic MA is a constructive theory.

Theorem 19. *If T is a constructive arithmetic theory, then*

$$T \leq_1 \mathcal{L}(T).$$

We see that predicate logic of any constructive arithmetic theory T is not simpler then the theory T itself. It can easily be checked that the set R of realizable arithmetic sentences is a constructive arithmetic theory. As the set R is not arithmetical, this imply the nonarithmeticity of the predicate $Rw(x)$ mentioned above. By Theorem 7, we obtain the nonarithmeticity of the predicate $Ri(x)$. The predicates $Re(x)$ and $Ru(x)$ can not be considered in the framework of the logics of constructive theories because in their definition a specific character of recursive realizability is used.

Theorem 19 can be extended to a wider class of arithmetic theories. Let A and B be arithmetical Σ_1-formulas with the only parameter x. Let $IS(A, B)$ denote the formula

$$\forall x \neg (A \,\&\, B) \,\&\, \neg \forall x \, \exists y \, ((A \to y = 0) \,\&\, (B \to y \neq 0)).$$

There exist formulas A and B such that the theory

$$\mathsf{HAIS}(A, B) = \mathsf{HA} + IS(A, B)$$

is consistent relative to HA.

An arithmetic theory T will be called *an intuitionistic IS-theory* iff T is closed under deducibility in HA and there are formulas A and B such that $IS(A, B) \in T$.
An arithmetic formula is called negative if it does not contain $\vee$ and $\exists$. Let T^- denote the negative fragment of an arithmetic theory T, i. e., the set of negative sentences in T.
Theorem 20. *If T is an intuitionistic arithmetic IS-theory, then $T^- \leq_1 \mathcal{L}(T)$.*

This theorem implies the following proposition.
Theorem 21. *If an interpretation* M *of the language LA^ω is such that a computable function is associated with any object of type* (0 $\to$ 0) *and the axioms of the system* HA *are* M*-realizable, then the logic $\mathcal{L}(MR(\mathrm{M}))$ is not arithmetical.*

We say that *Markov's principle holds* in an arithmetic theory T iff every one of the arithmetic instances of the predicate formula M is in T.

Theorem 22. *For any intuitionistic arithmetic IS-theory T, if Markov's principle holds in T, then $T \leq_1 \mathcal{L}(T)$.*

The proof of this theorem consists of the description of an algorithm that transforms any arithmetic sentence Φ into a predicate formula Φ' such that $\Phi \in T$ iff the formula Φ' is T-valid. This construction admits the following generalization useful from the technical point of view.

For any arithmetic theory T the notion of a T-valid scheme is defined in a natural way. Let T be an intuitionistic arithmetic IS-theory. Then the following theorem on schemes holds.

Theorem 23. *For any scheme $\mathfrak{A}$ one can effectively construct a predicate formula $\mathfrak{A}^*$ such that $\mathfrak{A}^* \in \mathcal{L}(T)$ iff the scheme $\mathfrak{A}$ is T-valid.*

The theorems above give a lower bound for complexity of the predicate logics of recursive realizability and modified realizability based on a constructive interpretation of the language LA^ω: all of them are not arithmetical. In some cases, one can obtain a more precise characterization of predicate logics of constructive arithmetic theories.

Following Vardanyan, we say that an arithmetic formula $\Phi(x, y)$ *is decidable* in an arithmetic theory T iff for any $m, n \in \mathbb{N}$, one of the formulas $\Phi(m, n)$ and $\neg\Phi(m, n)$ is in T. A set $A \subseteq \mathbb{N}$ is enumerable in a theory T iff there exists a decidable in T formula $\Phi(x, y)$ such that $A = \{m | \exists n\, [\Phi(m, n) \in T]\}$. A theory T is called *internally enumerable* iff the set of the Gödel numbers of the sentences in T is enumerable in T. In particular, every enumerable extension of PA or HA is internally enumerable. An arithmetic theory T is said *to have the existential property* iff for any formula of the form $\exists x\, \Phi(x)$ in T there exists a number n such that $\Phi(n) \in T$. HA and the theories based on constructive semantics like recursive realizability, modified realizability or Gödel's interpretation have the existential property.

Theorem 24. *Suppose an intuitionistic IS-theory T is internally enumerable and has the existential property. Then the logic $\mathcal{L}(T)$ is Π_1^T-complete.*

This theorem has the following applications.

Theorem 25. *The predicate logic of the Markov arithmetic* MA *is* Π_2*-complete.*

Theorem 26. *The predicate logic* $\mathcal{L}(D(\text{HRO}))$ *based on the Gödel interpretation and the system of hereditary recursive operations* HRO *is not enumerable.*

Consider another semantics based on the Gödel interpretation and the system HRO. The system HRO admits arithmetization in the sense, that for any type σ there is an arithmetic formula $V_\sigma(x)$ defining the set HRO_σ, thus the formula Φ^D can be «translated» into an arithmetic formula $(\Phi^D)^\circ$. Let $PD(\text{HRO})$ denote the set of arithmetic sentences Φ such that $\text{HA} \vdash (\Phi^D)^\circ$.

Theorem 27. *The predicate logic* $\mathcal{L}(PD(\text{HRO}))$ *is* Π_2*-complete.*

Internally enumerable theories usually are not complete. It is of interest to obtain upper bounds for arithmetic complexity of predicate logics of complete constructive arithmetic theories. This problem is parially solved by the following theorem.

Theorem 28. *Suppose that an intuitionistic arithmetic theory* T *is complete, has the disjunction property, and theorem on schemes holds for* T*. Then the logic* $\mathcal{L}(T)$ *is* Π_1^T*-complete.*

This theorem makes possible to obtain a precise characterization of the logic of recursive realizability. Let V be the set of (the Gödel numbers of) true arithmetic sentences.

Theorem 29. *The predicate logic of recursive realizability* $\mathcal{L}(R)$ *is* Π_1^V*-complete.*

It is known that for any A, every Π_1^A-complete set is recursively isomorphic to $\overline{A'}$, i. e., the complement of A', where A' is the jump of A. Thus Theorem 29 means that $\mathcal{L}(R)$ is recursively isomorphic to $\overline{V'}$. On the other hand, V is recursively isomorphic to $\emptyset^{(\omega+1)}$. Therefore the following proposition holds.

Theorem 30. *The predicate logic of recursive realizability $\mathcal{L}(R)$ is recursively isomorphic to the set $\overline{\emptyset^{(\omega+1)}}$.*

$\mathcal{L}(R)$ is the class of weakly realizable predicate formulas, that is recursively isomorphic to the set of irrefutable formulas. Thus the following proposition holds.

Theorem 31. *The sets of weakly realizable predicate formulas and of irrefutable predicate formulas are recursively isomorphic to the set $\overline{\emptyset^{(\omega+1)}}$.*

Primitive recursive realizability

It is of interest to consider variants of intuitionistic semantics, in which not the entire class of partial recursive functions is used for the interpretation of effective operations, as in Kleene's recursive realizability, but some of its subclasses. In 1994 Damnjanović introduced the concept of strictly primitive recursive realizability for arithmetic formulas, which combines the ideas of recursive realizability and Kripke models. In 2003 my Ph. D. student Park proved that the predicate logic of strictly primitive recursive realizability is nonarithmetical. The proof was essentially based on the claim from Damnjanović's paper that the calculus IQC is sound with respect to strictly primitive recursive realizability. Later, it was proved by me that this claim is erroneous. However, the result of Park remains true.

Theorem 32. *The predicate logic of strictly primitive recursive realizability is not arithmetical.*

Another variant of primitive recursive realizability was proposed by Salehi in 2000. The following theorem was proved by me.
Theorem 33. *There is a closed arithmetic formula wich is strictly primitive recursively realizable but is not primitive recursively realizable by Salehi.*

Of course, the negation of the formula from Theorem 33 is primitive recursively realizable by Salehi but is not strictly primitive recursively realizable.

The nonarithmeticity of the predicate logic of primitive recursive realizability by Salehi was proved by another my Ph. D. student Viter. His technically complex proof is based on the mentioned above results about predicate logics of constructive theories and results of Ardeshir on the translation of intuitionistic predicate logic into basic predicate logic. Recently I have published another, conceptually and technically simpler proof of the same result.

THE END