

Program Logics of Renominative Level with the Composition of Predicate Complement

Mykola Nikitchenko^[0000-0002-4078-1062], Oksana Shkilniak^[0000-0003-4139-2525],
Stepan Shkilniak^[0000-0001-8624-5778]

Taras Shevchenko National University of Kyiv,
60 Volodymyrska Street, City of Kyiv, Ukraine, 01033
mykola.nikitchenko@gmail.com, me.oksana@gmail.com,
sssh@unicyb.kiev.ua

Abstract. Program logics are widely used for software verification. Such logics are based on formal program models and reflect main program properties. Among various program logics, Floyd-Hoare logic and its variants take a special place because of its naturalness and simplicity. But such logics are oriented on total pre- and post-conditions, and in the case of partial conditions they become unsound. Different methods to overcome this problem were proposed in our previous works. One of the methods involves extension of program algebras with the composition of predicate complement. This permits to modify rules of the logic making them sound. Such modification requires introduction of undefinedness conditions into logic rules. In this paper we continue our research of such logics. We investigate a special predicate logic called logic of renominative (quantifier-free) level with the composition of predicate complement. This logic is a constituent part of the program logic. We introduce a special consequence relation for this logic, construct a sequent calculus, and prove its soundness and completeness.

Keywords: software verification, program logic, Floyd-Hoare logic, partial predicate, soundness, completeness.

1 Introduction

The formalism of program logics is the main instrument for software verification [1]. To be effective, such logics should reflect main program properties. Therefore, adequate formal program models should be constructed which will form a base for a program logic. Among such logics we should point to Floyd-Hoare logic and its variants as quite natural and simple [2, 3]. But such logics are oriented on total pre- and post-conditions, and in the case of partial conditions (predicates) they become unsound.

In our previous works [4, 5] we considered several methods to extend Floyd-Hoare logic for partial predicates, in particular, we proposed two methods: 1) introduction of special rule constraints; and 2) restriction of the class of program assertions (of Hoare triples). Both methods make a logic sound but they are difficult for practical usage or

are rather restrictive. Here we study one more method which proposes to extend program algebras with the composition of predicate complement [6, 7]. Introduction of this composition permits to modify rules in such a way that they become sound, but a negative side of this proposal is that logic becomes more complicated. In this case, undefinedness conditions for predicates should be taken into account.

In this paper we continue our research of logics with the composition of predicate complement. We concentrate on a base logic which is a constituent part of program logic. This logic is a special logic of partial quasiary predicates of renominative (quantifier-free) level. We introduce a consequence relation with undefinedness conditions, study its properties, and define a sequent calculus. We prove the soundness and completeness theorems for this logic with the composition of predicate complement.

2 Program Algebras with the Composition of Predicate Complement

According to the principles of *composition-nominative approach* [8, 9] we construct program logics based on program algebras. Such algebras are defined in the following way [9, 10]:

- 1) a set D of *data* processed by programs is defined;
- 2) the classes of *partial predicates* $Pr = D \xrightarrow{p} Bool$ and *partial functions* $Fn = D \xrightarrow{p} D$ are defined;
- 3) *operations (compositions)* over Pr and Fn are specified.

This scheme leads to *two-sorted program algebras*. In our previous works we considered program algebras with traditional compositions. But the problem of defining sound rules for program logics requires new compositions. Therefore, here we consider a program algebra extended with the *composition of predicate complement*. This unary predicate composition is defined in the following way ($p \in Pr, d \in D$):

$$(\square p)(d) = \begin{cases} T, & \text{if } p(d) \text{ is undefined,} \\ \text{undefined,} & \text{if } p(d) \text{ is defined.} \end{cases}$$

Specifying D as the class $D_{CC}(V, A)$ of *hierarchical nominative data* [7, 11] with complex names and values built over the set of basic names V and the set of basic values A , we can define a *complemented program algebra* as a two-sorted algebra [7]

$$\begin{aligned} CPAND_{CC}(V, A) &= (Pr_{CC}(V, A), Fn_{CC}(V, A); \\ AS^u, id, IF, WH, S_F^{\bar{u}}, S_P^{\bar{u}}, \Rightarrow v, v \Rightarrow_a, \vee, \neg, \exists x, \square) \end{aligned}$$

where $Pr_{CC}(V, A)$ and $Fn_{CC}(V, A)$ are classes of partial predicates and partial function over $D_{CC}(V, A)$ respectively; $AS^u, id, IF, WH, S_P^{\bar{u}}, S_F^{\bar{u}}$ are compositions of assignment, identity, conditional, cycle, superposition into predicate, superposition into function respectively; $\Rightarrow v$ and $v \Rightarrow_a$ are naming and denaming functions; $\vee, \neg, \exists x, \square$ are composition of disjunction, negation, existential quantification, and predicate complement; $v, u, x \in V^+$ are complex names, $\bar{u} \in \bar{U}$ is a sequence of

complex names. This algebra is quite expressive to present formal semantics of rather complex programs.

A special program logic of Floyd-Hoare type based on such algebras is presented in [7]. Its distinctive feature is introduction of new rules which are sound for partial predicates and which use preconditions constructed with the help of the composition of predicate complement.

For example, a classical rule of Floyd-Hoare logic for sequential execution of operators f and g has the form

$$R_SEQ \frac{\{p\}f\{q\}, \{q\}g\{r\}}{\{p\}f \bullet g\{r\}}$$

where $f \bullet g$ denotes sequential execution of f and g .

This rule is not sound in the case of partial predicates [4]. Therefore, a new sound rule based on extended program algebra was introduced [7]:

$$R_SSEQ \frac{\{p\}f\{q\}, \{q\}g\{r\}, \{\bar{q}\}g\{r\}}{\{p\}f \bullet g\{r\}}.$$

Obtained program logic can be an important instrument of program verification. So, its thorough investigation is required. This is a rather complicated challenge; therefore, we start with more simple logics. First, we identify a special predicate logic as a constituent part of the program logic. Such predicate logic can be considered as a logic defining constraints (program annotations). Second, we will consider here only logic L^{QCR} of renominative level which can be characterized as quantifier-free predicate logic of partial quasiary predicates with the composition of predicate complement. The case of first-order logic with quantifiers and functions is planned to study in the forthcoming papers.

3 Logic of Partial Quasiary Predicates of Renominative Level with the Composition of Predicate Complement

To define a logic L^{QCR} we should define [9, 10]

- its class of algebras;
- its language (based on logic signature);
- its class of interpretations;
- its consequence relation;
- its inference relation based on some calculus.

Formal definitions will be given in the next section. We will use the following notations:

- $S \xrightarrow{p} S'$ ($S \xrightarrow{t} S'$) is the class of partial (total) mappings from S to S' ;
- $p(d) \downarrow$ ($p(d) \uparrow$) means that p is defined (undefined) on d ;
- $p(d) \downarrow = T$ ($p(d) \downarrow = F$) means that p is defined on d with value T (F). For this case we also use simpler notation $p(d) = T$ ($p(d) = F$).

The terms and notations, not defined here, are treated in the sense of [12].

3.1 Complemented Algebras of Partial Quasiary Predicates of Renominative Level

Let V be a set of *names* (variables) and A be a set of *values*. The class ${}^V A$ of *nominative sets* (partial assignments, partial data) is defined as the class of all partial mappings from V to A , thus, ${}^V A = V \xrightarrow{p} A$.

Nominative sets represent *states of program variables*.

The main operation for nominative sets is a total unary parametric *renomination* $r_{x_1, \dots, x_n}^{v_1, \dots, v_n} : {}^V A \xrightarrow{t} {}^V A$, where $v_1, \dots, v_n, x_1, \dots, x_n$ are names, and v_1, \dots, v_n are distinct [12]. Intuitively, given nominative set d this operation yields a new nominative set changing the values of v_1, \dots, v_n to the values of x_1, \dots, x_n respectively. We also use simpler notation for this renomination: $r_{\bar{x}}$. We write $x \in \bar{v}$ to denote that x is a variable from \bar{v} ; we write $\bar{v} \cup \bar{x}$ to denote the set of variables that occur in the sequences \bar{v} and \bar{x} .

The set of *assigned variables* (names) in d is denoted $asn(d)$.

Let $Pr_A^V = {}^V A \xrightarrow{p} Bool$ be the set of all *partial predicates* over ${}^V A$. Such predicates are called *partial quasiary predicates*. For a predicate $p \in Pr_A^V$ its *truth*, *falsity*, and *undefinedness domains* are denoted $T(p)$, $F(p)$, and $\perp(p)$ respectively. Please note that these domains do not intersect pairwise and their union is equal to ${}^V A$; thus, predicate p is defined by $T(p)$ and $F(p)$ only, because $\perp(p) = {}^V A \setminus (T(p) \cup F(p))$.

Operations over Pr_A^V are called *compositions*. Basic compositions of renominative level over quasiary predicates are *disjunction* \vee , *negation* \neg , and *renomination* $R_{\bar{x}}^{\bar{v}}$.

We extend this set with the composition of *predicate complement* \square .

These compositions are defined by the following formulas ($p, q \in Pr_A^V$):

- $T(p \vee q) = T(p) \cup T(q)$; $F(p \vee q) = F(p) \cap F(q)$;
- $T(\neg p) = F(p)$; $F(\neg p) = T(p)$;
- $T(R_{\bar{x}}^{\bar{v}}(p)) = \{d \in {}^V A \mid r_{\bar{x}}^{\bar{v}}(d) \in T(p)\}$; $F(R_{\bar{x}}^{\bar{v}}(p)) = \{d \in {}^V A \mid r_{\bar{x}}^{\bar{v}}(d) \in F(p)\}$;
- $T(\square p) = \perp(p)$; $F(\square p) = \emptyset$.

Please note that definitions of disjunction and negation are similar to *strong Kleene's connectives* [13]. We consider \square as a composition of propositional level.

A tuple

$$A^{QCR}(V, A) = \langle Pr_A^V; \neg, \vee, R_{\bar{x}}^{\bar{v}}, \square \rangle$$

is called a *complemented algebra of partial quasiary predicates of renominative level*.

A class of such algebras (with different A) forms a semantic base for a logic L^{QCR} .

Now we describe the main properties of $A^{QCR}(V, A)$. We do not formulate traditional properties of propositional compositions of disjunction and negation [9, 14], but concentrate on properties of compositions of renomination and complement.

Compositions of disjunction and negation have traditional properties; in contrast to these compositions, the composition of predicate complement is more complicated: it does not have the monotonicity property and it does not have distributivity properties with respect to disjunction. For this composition we identify the following properties.

Lemma 1. For any $p \in Pr_A^V$ we have

$$\Box \neg p = \Box p ; \Box \Box p = \Box p ; \Box \Box \Box p = \Box \Box p .$$

Lemma 2. For any $p \in Pr_A^V$ we have

- 1) $T(\neg \Box p) = \emptyset ; F(\neg \Box p) = T(\Box p) ; \perp(\neg \Box p) = \perp(\Box p) ;$
- 2) $T(R_{\bar{x}}^{\bar{v}}(\Box p)) = T(\Box(R_{\bar{x}}^{\bar{v}}(p))) ; F(R_{\bar{x}}^{\bar{v}}(\Box p)) = \emptyset ; \perp(R_{\bar{x}}^{\bar{v}}(\Box p)) = \perp(\Box(R_{\bar{x}}^{\bar{v}}(p))) .$

The notion of unessential variable is important for the composition of renomination. A name (variable) z is *unessential* for predicate $p \in Pr_A^V$, if for any $d \in {}^V A$ the value of p does not depends on the value of z [9, 12].

Lemma 3. The following properties of the compositions of renomination and predicate complement hold for any $p \in Pr_A^V$:

- $R\vee$) $R_{\bar{x}}^{\bar{v}}(p \vee q) = R_{\bar{x}}^{\bar{v}}(p) \vee R_{\bar{x}}^{\bar{v}}(q) ;$
- $R\neg$) $R_{\bar{x}}^{\bar{v}}(\neg p) = \neg R_{\bar{x}}^{\bar{v}}(p) ;$
- RR) $R_{\bar{x}}^{\bar{v}}(R_{\bar{y}}^{\bar{w}}(p)) = R_{\bar{x}}^{\bar{v}} \circ_{\bar{y}}^{\bar{w}}(p) ;$
- $R\Box$) $R_{\bar{x}}^{\bar{v}}(\Box p) = \Box R_{\bar{x}}^{\bar{v}}(p) ;$
- R) $R(p) = p ;$
- RI) $R_{z,\bar{x}}^{z,\bar{v}}(p) = R_{\bar{x}}^{\bar{v}}(p) ;$
- RU) $R_{y,\bar{x}}^{z,\bar{v}}(p) = R_{\bar{x}}^{\bar{v}}(p)$ if $z \in V$ is unessential for p .

3.2 Language (signature and formulas) of L^{QCR}

Let Ps be a set of *predicate symbols*, V be an infinite set of *names (variables)*. Usually, within V a subset U of *unessential variables* is identified but here we will not go into detail [12]. A tuple

$$\Sigma^{QCR} = (V, U; \vee, \neg, R_{\bar{x}}^{\bar{v}}, \Box; Ps)$$

is called the *language signature*.

For simplicity, we use the same notation for symbols of compositions and compositions themselves.

Given Σ^{QCR} , we define inductively the *language* of L^{QCR} – the *set of formulas* denoted $Fr(L^{QCR})$ or simply Fr :

- if $P \in Ps$ then $P \in Fr$. Formulas of such forms are called *atomic*;
- if $\Phi, \Psi \in Fr$ then $\Phi \vee \Psi, \neg \Phi, R_{\bar{x}}^{\bar{v}} \Phi, \Box \Phi \in Fr$.

3.3 L^{QCR} -interpretations

Let $A^{QCR}(V, A) = \langle Pr_A^V; \neg, \vee, R_{\bar{x}}^{\bar{v}}, \square \rangle$ be a complemented algebra of a signature $\Sigma^{QCR} = (V, U; \vee, \neg, R_{\bar{x}}^{\bar{v}}, \square; Ps)$, $I_Q^{Ps} = Ps \xrightarrow{t} Pr_A^V$ be an *interpretation mapping of predicate symbols*. Then a tuple $J(\Sigma^{QCR}) = (A^{QCR}(V, A), I_Q^{Ps})$ is called an L^{QCR} -interpretation.

We simplify notation for L^{QCR} -interpretation $J(\Sigma^{QCR})$ omitting L^{QCR} and Σ^{QCR} .

In interpretation J , an algebra $A^{QCR}(V, A)$ defines interpretations of composition symbols while I_Q^{Ps} defines interpretations of predicate symbols.

For given interpretation J and formula Φ , we can define by induction on the structure of Φ its value in J . Obtained predicate is denoted Φ_J .

Lemma 4. Let J be an interpretation and $\Phi, \Psi \in Fr$. Then

$$\begin{aligned} R) & R(\Phi)_J = \Phi_J; \\ RI) & R_{z, \bar{x}}^{z, \bar{v}}(\Phi)_J = R_{\bar{x}}^{\bar{v}}(\Phi)_J; \\ RU) & R_{y, \bar{x}}^{z, \bar{v}}(\Phi)_J = R_{\bar{x}}^{\bar{v}}(\Phi)_J \text{ if } z \in V \text{ is unessential for } \Phi; \\ R\neg) & R_{\bar{x}}^{\bar{v}}(\neg\Phi)_J = \neg R_{\bar{x}}^{\bar{v}}(\Phi)_J; \\ R\vee) & R_{\bar{x}}^{\bar{v}}(\Phi \vee \Psi)_J = R_{\bar{x}}^{\bar{v}}(\Phi)_J \vee R_{\bar{x}}^{\bar{v}}(\Psi)_J; \\ RR) & R_{\bar{x}}^{\bar{v}}(R_{\bar{y}}^{\bar{w}}(\Phi))_J = R_{\bar{x}}^{\bar{v}} \circ_{\bar{y}}^{\bar{w}}(\Phi)_J; \\ R\square) & R_{\bar{x}}^{\bar{v}}(\square\Phi)_J = \square R_{\bar{x}}^{\bar{v}}(\Phi)_J. \end{aligned}$$

3.4 Logical Consequence Relation under Conditions of Undefinedness

Introduction of composition \square requires more complicated consequence relation because undefinedness domains should be taken into consideration. Here we introduce new consequence relation between sets of formulas denoted $|\models_{IR}^\perp$ which generalizes irrefutability relation $|\models_{IR}$ [7].

Let $\Sigma \subseteq Fr$ and J be an interpretation.

We denote:

$$\bigcap_{\Phi \in \Sigma} T(\Phi_J) \text{ as } T^\wedge(\Sigma_J), \bigcap_{\Phi \in \Sigma} F(\Phi_J) \text{ as } F^\wedge(\Sigma_J), \bigcap_{\Phi \in \Sigma} \perp(\Phi_J) \text{ as } \perp^\wedge(\Sigma_J).$$

Here Σ_J denotes set $\{\Phi_J \mid \Phi \in \Sigma\}$.

Set Σ can be empty. In this case

$$T^\wedge(\Sigma) = T^\wedge(\emptyset) = F^\wedge(\Sigma) = F^\wedge(\emptyset) = \perp^\wedge(\Sigma) = \perp^\wedge(\emptyset) = {}^V A.$$

Let $\Gamma, U, \Delta \subseteq Fr$. Informally, the statement “ Δ is irrefutable consequence of Γ under undefinedness conditions U in interpretation J ” means

“for any $d \in {}^V A$ if $\varphi_J(d) \uparrow$ for any $\varphi \in U$ then it is not possible that $(\xi_J(d) = T$ for any $\xi \in \Gamma$ and $\psi_J(d) = F$ for any $\psi \in \Delta$)”.

This statement is equivalent to the following statement:

“for any $d \in {}^V A$ if $d \in \perp^\wedge(U_j)$ then it is not possible that $d \in T^\wedge(\Gamma_j) \cap F^\wedge(\Delta_j)$ ”.

The former statement can be reformulated as follows:

“for any $d \in {}^V A$ it is not possible that $(d \in \perp^\wedge(U_j))$ and $d \in T^\wedge(\Gamma_j) \cap F^\wedge(\Delta_j)$ ”.

Finally, we obtain the following statement:

$$\perp^\wedge(U_j) \cap T^\wedge(\Gamma_j) \cap F^\wedge(\Delta_j) = \emptyset$$

So, we come to the following formal definition: Δ is *irrefutable consequence* of Γ under undefinedness conditions U in interpretation J (denoted $U / \Gamma \vdash_{IR}^\perp \Delta$) if

$$T^\wedge(\Gamma_j) \cap \perp^\wedge(U_j) \cap F^\wedge(\Delta_j) = \emptyset.$$

In particular, for $U = \emptyset$ we obtain *irrefutability consequence relation* $\Gamma \vdash_{IR} \Delta$.

Δ is *logical irrefutability consequence* of Γ under undefinedness conditions U (denoted $U / \Gamma \vdash_{IR}^\perp \Delta$), if $U / \Gamma \vdash_{IR}^\perp \Delta$ for any interpretation J .

In particular, for $U = \emptyset$, we get traditional *logical irrefutability relation* $\Gamma \vdash_{IR} \Delta$.

Let us now describe the main properties of the consequence relation \vdash_{IR}^\perp for propositional level.

By definition of \vdash_{IR}^\perp , we obtain *monotonicity*:

M) Let $\Gamma \subseteq \Lambda$, $U \subseteq W$, and $\Delta \subseteq \Sigma$; then $U / \Gamma \vdash_{IR}^\perp \Delta \Rightarrow W / \Lambda \vdash_{IR}^\perp \Sigma$.

The following properties describe conditions under which \vdash_{IR}^\perp holds.

Theorem 1. For any $U, \Gamma, \Delta \subseteq Fr$, $\Phi \in Fr$:

- C) $U / \Phi, \Gamma \vdash_{IR}^\perp \Delta, \Phi$;
- C_{UL}) $U, \Phi / \Phi, \Gamma \vdash_{IR}^\perp \Delta$;
- C_{UR}) $U, \Phi / \Gamma \vdash_{IR}^\perp \Delta, \Phi$;
- C_{\square}) $U / \Gamma \vdash_{IR}^\perp \Delta, \square \Phi \square$.

Proof. Property C holds because $T(\Phi_j) \cap F(\Phi_j) = \emptyset$.

For property C_{UL} we take into consideration that $\perp(\Phi_j) \cap T(\Phi_j) = \emptyset$.

For property C_{UR} we take into consideration that $\perp(\Phi_j) \cap F(\Phi_j) = \emptyset$.

Property C_{\square} holds because $F(\square \Phi_j) = \emptyset$.

For \vdash_{IR}^\perp the following properties of formula decomposition hold.

Theorem 2. For any $U, \Gamma, \Delta \subseteq Fr$, $\Phi, \Psi, \Theta \in Fr$:

- \neg_L) $U / \neg \Phi, \Gamma \vdash_{IR}^\perp \Delta \Leftrightarrow U / \Gamma \vdash_{IR}^\perp \Delta, \Phi$;
- \neg_R) $U / \Gamma \vdash_{IR}^\perp \Delta, \neg \Phi \Leftrightarrow U / \Phi, \Gamma \vdash_{IR}^\perp \Delta$;
- \vee_L) $U / \Phi \vee \Psi, \Gamma \vdash_{IR}^\perp \Delta \Leftrightarrow U / \Phi, \Gamma \vdash_{IR}^\perp \Delta$ and $U / \Psi, \Gamma \vdash_{IR}^\perp \Delta$;
- \vee_R) $U / \Gamma \vdash_{IR}^\perp \Delta, \Phi \vee \Psi \Leftrightarrow U / \Gamma \vdash_{IR}^\perp \Delta, \Phi, \Psi$;
- \neg_U) $U, \neg \Theta / \Gamma \vdash_{IR}^\perp \Delta \Leftrightarrow U, \Theta / \Gamma \vdash_{IR}^\perp \Delta$;
- \vee_U) $U, \Phi \vee \Theta / \Gamma \vdash_{IR}^\perp \Delta \Leftrightarrow U, \Phi, \Theta / \Gamma \vdash_{IR}^\perp \Delta$ and $U, \Phi / \Gamma \vdash_{IR}^\perp \Theta, \Delta$ and $U, \Theta / \Gamma \vdash_{IR}^\perp \Phi, \Delta$;
- \square_U) $U, \square \Phi \square / \Gamma \vdash_{IR}^\perp \Delta \Leftrightarrow U / \Phi, \Gamma \vdash_{IR}^\perp \Delta$ and $U / \Gamma \vdash_{IR}^\perp \Delta, \Phi$;
- \square_L) $U / \square \Phi, \Gamma \vdash_{IR}^\perp \Delta \Leftrightarrow U, \Phi / \Gamma \vdash_{IR}^\perp \Delta$.

Proof. Property \neg_U holds because $\perp(\neg \Phi_j) = \perp(\Phi_j)$.

Property \vee_U holds because

$$\perp(\Phi_J \vee \Psi_J) = (\perp(\Phi_J) \wedge \perp(\Psi_J)) \cup (\perp(\Phi_J) \wedge F(\Psi_J)) \cup (F(\Phi_J) \wedge \perp(\Psi_J)).$$

Property \Box_U holds because $\perp(\Box \Phi_J) \equiv T(\Phi_J) \cup F(\Phi_J)$.

Properties $\neg_L, \neg_R, \vee_L, \vee_R$ are similar to properties of \models_{IR} [9, 12, 13]. Properties $\neg_U, \vee_U, \Box_L, \Box_U$ are special for L^{QCR} .

Let us consider properties of relation \models_{IR}^U of a renominative level. Their proofs are based on Theorem 2. Each of properties $R, RI, RU, RR, R\neg, R\vee, R\Box$ of Lemma 4 induces three corresponding properties for \models_{IR}^U , depending on the position of a formula (in the left side of \models_{IR}^U , in the right side of \models_{IR}^U , in the undefinedness conditions of \models_{IR}^U). Such properties are formulated in a similar way, for example, the following properties $R\Box_L, R\Box_R, R\Box_U$ are induced by $R\Box$:

$$R\Box_L) U / R_x^{\bar{\vee}}(\Box \Phi), \Gamma \models_{IR}^{\perp} \Delta \Leftrightarrow U / \Box R_x^{\bar{\vee}}(\Phi), \Gamma \models_{IR}^{\perp} \Delta;$$

$$R\Box_R) U / \Gamma \models_{IR}^{\perp} \Delta, R_x^{\bar{\vee}}(\Box \Phi) \Leftrightarrow U / \Gamma \models_{IR}^{\perp} \Delta, \Box R_x^{\bar{\vee}}(\Phi);$$

$$R\Box_U) U, R_x^{\bar{\vee}}(\Box \Phi) / \Gamma \models_{IR}^{\perp} \Delta \Leftrightarrow U, \Box R_x^{\bar{\vee}}(\Phi) / \Gamma \models_{IR}^{\perp} \Delta.$$

4 Sequent Calculus for L^{QCR}

Usually, inference relations are defined by some axiomatic systems (calculi). We present here a system that formalizes logical consequence relation between two sets of formulas. Such systems are called *sequent calculi*.

We construct a sequent calculus C^{QCR} for relation \models_{IR}^{\perp} .

The main objects of this calculus are *sequents*. Here we consider only the case with *finite* sequents. We construct calculus in the style of semantic tableau, so, we will treat sequents as finite sets of formulas signed (marked, indexed) by symbols \vdash, \dashv , and \perp .

Formulas from Γ (they are signed by \vdash) are called *T-formulas*, formulas from Δ (they are signed by \dashv) are called *F-formulas*, and formulas from U (they are signed by \perp) are called *\perp -formulas*.

Sequents are denoted $\vdash \Gamma \perp U \dashv \Delta$, in abbreviated form Σ .

The *derivation* in a sequent calculus has the form of a tree whose vertices are sequents. Such trees are called *sequent trees*.

The rules of sequent calculus are called *sequent forms*. They are syntactical analogs of the semantic properties of the corresponding relations of logical consequence. Details of the definition of *sequent tree* can be found in [12].

Closed sequents are axioms of the sequent calculus.

A *closed sequent* is specified in such a way that the following condition should hold:

$$\text{if sequent } \vdash \Gamma \perp U \dashv \Delta \text{ is closed then } U / \Gamma \models_{IR}^{\perp} \Delta.$$

Sequent calculus is defined by basic sequent forms and closure conditions of sequents.

For C^{QCR} we take the following closure conditions:

sequent $\vdash_{\Gamma} \perp U \rightarrow \Delta$ is closed if condition $C \vee C_{UL} \vee C_{UR} \vee C_{\square}$ holds.

Here $C, C_{UL}, C_{UR}, C_{\square}$ are the following basic closure conditions:

- C) exists Φ : $\Phi \in \Gamma$ and $\Phi \in \Delta$;
- C_{UL}) exists Φ : $\Phi \in \Gamma$ and $\Phi \in U$;
- C_{UR}) exists Φ : $\Phi \in \Delta$ and $\Phi \in U$;
- C_{\square}) exists Φ : $\square \Phi \in \Delta$.

Theorem 3. If sequent $\vdash_{\Gamma} \perp U \rightarrow \Delta$ is closed then $U/\Gamma \models_{IR}^{\perp} \Delta$.

Proof. The theorem statement follows directly from Theorem 1.

The sequent forms of decomposition of compositions \vee, \neg, \square are induced by the corresponding properties of formulas decomposition, in particular, basic sequent forms of C^{QCR} calculus are induced by the formula decomposition properties $\neg_L, \neg_R, \vee_L, \vee_R, \neg U, \vee U, \square U, \square L$:

$$\begin{array}{l} \vdash_{\neg} \frac{\neg \Phi, \Sigma}{\neg \neg \Phi, \Sigma}; \quad \neg \vdash_{\neg} \frac{\vdash \Phi, \Sigma}{\neg \neg \Phi, \Sigma}; \quad \perp \vdash_{\neg} \frac{\perp \Phi, \Sigma}{\perp \neg \Phi, \Sigma}; \\ \vdash_{\vee} \frac{\vdash \Phi, \Sigma \quad \vdash \Psi, \Sigma}{\vdash \Phi \vee \Psi, \Sigma}; \quad \neg \vdash_{\vee} \frac{\neg \Phi, \neg \Psi, \Sigma}{\neg \Phi \vee \Psi, \Sigma}; \quad \perp \vdash_{\vee} \frac{\perp \Phi, \perp \Psi, \Sigma \quad \perp \Phi, \neg \Psi, \Sigma \quad \neg \Phi, \perp \Psi, \Sigma}{\perp \Phi \vee \Psi, \Sigma}; \\ \perp \vdash_{\square} \frac{\vdash \Phi, \Sigma \quad \neg \Phi, \Sigma}{\perp \square \Phi, \Sigma}; \quad \vdash_{\square} \frac{\perp \Phi, \Sigma}{\vdash \square \Phi, \Sigma}. \end{array}$$

For the composition of renomination we use the following forms of equivalent transformations:

$$\begin{array}{l} \vdash_{R\neg} \frac{\vdash \neg R_x^{\bar{v}}(\Phi), \Sigma}{\vdash R_x^{\bar{v}}(\neg \Phi), \Sigma}; \quad \neg \vdash_{R\neg} \frac{\neg \neg R_x^{\bar{v}}(\Phi), \Sigma}{\neg R_x^{\bar{v}}(\neg \Phi), \Sigma}; \quad \perp \vdash_{R\neg} \frac{\perp \neg R_x^{\bar{v}}(\Phi), \Sigma}{\perp R_x^{\bar{v}}(\neg \Phi), \Sigma}; \\ \vdash_{R\vee} \frac{\vdash R_x^{\bar{v}}(\Phi) \vee R_x^{\bar{v}}(\Psi), \Sigma}{\vdash R_x^{\bar{v}}(\Phi \vee \Psi), \Sigma}; \quad \neg \vdash_{R\vee} \frac{\neg R_x^{\bar{v}}(\Phi) \vee R_x^{\bar{v}}(\Psi), \Sigma}{\neg R_x^{\bar{v}}(\Phi \vee \Psi), \Sigma}; \quad \perp \vdash_{R\vee} \frac{\perp R_x^{\bar{v}}(\Phi) \vee R_x^{\bar{v}}(\Psi), \Sigma}{\perp R_x^{\bar{v}}(\Phi \vee \Psi), \Sigma}; \\ \vdash_{R\square} \frac{\vdash \square R_x^{\bar{v}}(\Phi), \Sigma}{\vdash R_x^{\bar{v}}(\square \Phi), \Sigma}; \quad \neg \vdash_{R\square} \frac{\neg \square R_x^{\bar{v}}(\Phi), \Sigma}{\neg R_x^{\bar{v}}(\square \Phi), \Sigma}; \quad \perp \vdash_{R\square} \frac{\perp \square R_x^{\bar{v}}(\Phi), \Sigma}{\perp R_x^{\bar{v}}(\square \Phi), \Sigma}; \\ \vdash_{RR} \frac{\vdash R_x^{\bar{v}} \circ_y^{\bar{w}}(\Phi), \Sigma}{\vdash R_x^{\bar{v}}(R_y^{\bar{w}}(\Phi)), \Sigma}; \quad \neg \vdash_{RR} \frac{\neg R_x^{\bar{v}} \circ_y^{\bar{w}}(\Phi), \Sigma}{\neg R_x^{\bar{v}}(R_y^{\bar{w}}(\Phi)), \Sigma}; \quad \perp \vdash_{RR} \frac{\perp R_x^{\bar{v}} \circ_y^{\bar{w}}(\Phi), \Sigma}{\perp R_x^{\bar{v}}(R_y^{\bar{w}}(\Phi)), \Sigma}. \end{array}$$

Here $R_x^{\bar{v}} \circ_y^{\bar{w}}(\Phi)$ represents application of two successive renominations $R_x^{\bar{v}}(R_y^{\bar{w}}(\Phi))$ [12].

Forms of simplification:

$$\vdash_R \frac{\vdash \Phi, \Sigma}{\vdash R(\Phi), \Sigma}; \quad \neg \vdash_R \frac{\neg \Phi, \Sigma}{\neg R(\Phi), \Sigma}; \quad \perp \vdash_R \frac{\perp \Phi, \Sigma}{\perp R(\Phi), \Sigma};$$

$$\begin{array}{ccc}
\vdash_{\text{RI}} \frac{\vdash R_{\bar{x}}^{\bar{v}}(\Phi), \Sigma}{\vdash R_{z, \bar{x}}^{z, \bar{v}}(\Phi), \Sigma}; & \vdash_{\text{RI}} \frac{\neg R_{\bar{x}}^{\bar{v}}(\Phi), \Sigma}{\neg R_{z, \bar{x}}^{z, \bar{v}}(\Phi), \Sigma}; & \perp_{\text{RI}} \frac{\perp R_{\bar{x}}^{\bar{v}}(\Phi), \Sigma}{\perp R_{z, \bar{x}}^{z, \bar{v}}(\Phi), \Sigma}; \\
\vdash_{\text{RU}} \frac{\vdash R_{\bar{u}}^{\bar{v}}(\Phi), \Sigma}{\vdash R_{z, \bar{u}}^{y, \bar{v}}(\Phi), \Sigma}; & \vdash_{\text{RU}} \frac{\neg R_{\bar{u}}^{\bar{v}}(\Phi), \Sigma}{\neg R_{z, \bar{u}}^{y, \bar{v}}(\Phi), \Sigma}; & \perp_{\text{RU}} \frac{\perp R_{\bar{x}}^{\bar{v}}(\Phi), \Sigma}{\perp R_{y, \bar{x}}^{z, \bar{v}}(\Phi), \Sigma}.
\end{array}$$

The names of the sequent forms are consistent with the names of the properties of the decomposition of the formulas. Introduction of undefinedness formulas additionally leads to new sequent forms with three premises (rule $\perp\vee$).

For basic rules of C^{QCR} we have the following main properties.

Theorem 4.

1. Let $\frac{\vdash \Lambda_{\perp} W_{\perp} K}{\vdash \Gamma_{\perp} U_{\perp} \Delta}$ be basic sequent form. Then
 - a) $U / \Gamma \models_{IR}^{\perp} \Delta \Leftrightarrow W / \Lambda \models_{IR}^{\perp} K$;
 - b) $U / \Gamma \not\models_{IR}^{\perp} \Delta \Leftrightarrow W / \Lambda \not\models_{IR}^{\perp} K$.
2. Let $\frac{\vdash \Lambda_{\perp} W_{\perp} K \quad \vdash X_{\perp} V_{\perp} Z}{\vdash \Gamma_{\perp} U_{\perp} \Delta}$ be basic sequent form. Then
 - a) $U / \Gamma \models_{IR}^{\perp} \Delta \Leftrightarrow W / \Lambda \models_{IR}^{\perp} K$ and $V / X \models_{IR}^{\perp} Z$;
 - b) $U / \Gamma \not\models_{IR}^{\perp} \Delta \Leftrightarrow W / \Lambda \not\models_{IR}^{\perp} K$ or $V / X \not\models_{IR}^{\perp} Z$.
3. Let $\frac{\vdash \Lambda_{\perp} W_{\perp} K \quad \vdash X_{\perp} V_{\perp} Z \quad \vdash M_{\perp} Y_{\perp} N}{\vdash \Gamma_{\perp} U_{\perp} \Delta}$ be basic sequent form. Then
 - a) $U / \Gamma \models_{IR}^{\perp} \Delta \Leftrightarrow W / \Lambda \models_{IR}^{\perp} K$ and $V / X \models_{IR}^{\perp} Z$ and $Y / M \models_{IR}^{\perp} N$;
 - b) $U / \Gamma \not\models_{IR}^{\perp} \Delta \Leftrightarrow W / \Lambda \not\models_{IR}^{\perp} K$ or $V / X \not\models_{IR}^{\perp} Z$ or $Y / M \not\models_{IR}^{\perp} N$.

Proof. The proof of the theorem is obtained by set-theoretic methods using a formula specifying relation \models_{IR}^{\perp} .

5 Soundness and Completeness of C^{QCR}

Now we prove soundness and completeness theorems for C^{QCR} .

Theorem 5 (soundness). Let sequent $\vdash \Gamma_{\perp} U_{\perp} \Delta$ be derivable in C^{QCR} . Then $U / \Gamma \models_{IR}^{\perp} \Delta$.

Proof. If $\vdash \Gamma_{\perp} U_{\perp} \Delta$ is derivable then a finite closed tree was constructed. From this follows that for any leaf of this tree its sequent $\vdash \Lambda_{\perp} W_{\perp} K$ is closed. Thus, by Theorem 4, $W / \Lambda \models_{IR}^{\perp} K$ holds. Therefore, for the root of the tree (sequent $\vdash \Gamma_{\perp} U_{\perp} \Delta$) we have that $U / \Gamma \models_{IR}^{\perp} \Delta$ holds.

The completeness is traditionally proved on the basis of theorems of the existence of a counter-model for the set of formulas of a non-closed path in the sequent tree. In this case a method of *model sets* is used.

We apply this method to the C^{QCR} calculus.

Set H of signed formulas is a *model set* (*Hintikka's set*) for L^{QCR} if the following conditions hold:

Decomposition conditions:

HC_U) For any $\Phi \in Fr$ at most one of $\vdash \Phi$, $\neg \Phi$, $\perp \Phi$ can belong to H ;

HC_{\Box}) For any $\Phi \in Fr$ it is not possible that $\neg \Box \Phi \in H$;

$H_{\neg L}$) If $\vdash \neg \Phi \in H$, then $\neg \Phi \in H$;

$H_{\neg R}$) If $\neg \neg \Phi \in H$, then $\vdash \Phi \in H$;

$H_{\vee L}$) If $\vdash \Phi \vee \Psi \in H$, then $\vdash \Phi \in H$ or $\vdash \Psi \in H$;

$H_{\vee R}$) If $\neg \Phi \vee \Psi \in H$, then $\neg \Phi \in H$ and $\neg \Psi \in H$;

$H_{\neg U}$) If $\perp \neg \Phi \in H$, then $\perp \Phi \in H$;

$H_{\vee U}$) If $\perp \Phi \vee \vartheta \in H$, then $\perp \Phi \in H$ and $\perp \vartheta \in H$

or $\perp \Phi \in H$ and $\neg \vartheta \in H$ or $\neg \Phi \in H$ and $\perp \vartheta \in H$;

$H_{\Box L}$) If $\vdash \Box \Phi \in H$, then $\perp \Phi \in H$;

$H_{\Box U}$) If $\perp \Box \Phi \in H$, then $\vdash \Phi \in H$ or $\neg \Phi \in H$.

Conditions for the composition of renomination are formulated in a similar way, for example, sequent forms RI and $R\Box$ induce the following conditions:

HRI_L) $\vdash R_{z,\bar{x}}^{\exists,\forall}(\Phi) \in H \Rightarrow \vdash R_{\bar{x}}^{\forall}(\Phi) \in H$;

HRI_R) $\neg \vdash R_{z,\bar{x}}^{\exists,\forall}(\Phi) \in H \Rightarrow \neg \vdash R_{\bar{x}}^{\forall}(\Phi) \in H$;

HRI_U) $\perp R_{z,\bar{x}}^{\exists,\forall}(\Phi) \in H \Rightarrow \perp R_{\bar{x}}^{\forall}(\Phi) \in H$;

$HR_{\Box L}$) $\vdash R_{\bar{x}}^{\forall}(\Box \Phi) \in H \Rightarrow \vdash \Box R_{\bar{x}}^{\forall}(\Phi) \in H$;

$HR_{\Box R}$) $\neg \vdash R_{\bar{x}}^{\forall}(\Box \Phi) \in H \Rightarrow \neg \vdash \Box R_{\bar{x}}^{\forall}(\Phi) \in H$;

$HR_{\Box U}$) $\perp R_{\bar{x}}^{\forall}(\Box \Phi) \in H \Rightarrow \perp \Box R_{\bar{x}}^{\forall}(\Phi) \in H$.

In the same way conditions HR_L , HR_R , HR_U , HRU_L , HRU_R , HRU_U , HRR_L , HRR_R , HRR_U , $HR_{\neg L}$, $HR_{\neg R}$, $HR_{\neg U}$, $HR_{\vee L}$, $HR_{\vee R}$, HR_{\vee} can be formulated.

A set $H \subseteq Fr$ is called *satisfiable* if there exist a set A , an interpretation J , and a nominative set $\delta \in {}^V A$ such that

- $\vdash \Phi \in H \Rightarrow \Phi_A(\delta) \downarrow = T$;
- $\neg \Phi \in H \Rightarrow \Phi_A(\delta) \downarrow = F$;
- $\perp \Phi \in H \Rightarrow \Phi_A(\delta) \uparrow$.

A set H of signed formulas for which the above-written conditions hold is called H^{QCR} -model.

Theorem 6. Let H be H^{QCR} -model for L^{QCR} . Then H is satisfiable.

Proof. Given H^{QCR} -model H , we should construct a set A , an interpretation J , and a nominative set $\delta \in {}^V A$ that demonstrate satisfiability of H . These constructions are rather complicated due to undefinedness conditions therefore here we do not present the proof in all details but demonstrate only its main parts.

Let $W = nm(H)$ be a set of subject names (variables) that occur in H . Let a set A duplicates W and $\delta \in {}^V A$ be a nominative set such that $asn(\delta) = W$.

Let us prescribe values of basic predicates on nominative set δ and nominative sets of the form $r_{\bar{x}}^{\bar{v}}(\delta)$. To do this, we use notations $P_A(\delta) = T$, $P_A(\delta) = F$, and $P_A(\delta) \uparrow$ to prescribe the value of P on d in algebra $A^{QCR}(V, A)$ equal to T , equal to F , and to be undefined respectively:

- $\vdash P \in H \Rightarrow P_A(\delta) = T$;
- $\neg \vdash P \in H \Rightarrow P_A(\delta) = F$;
- $\perp \vdash P \in H \Rightarrow P_A(\delta) \uparrow$;
- $\vdash R_{\bar{x}}^{\bar{v}}(P) \in H \Rightarrow P_A(r_{\bar{x}}^{\bar{v}}(\delta)) = T$;
- $\neg \vdash R_{\bar{x}}^{\bar{v}}(P) \in H \Rightarrow P_A(r_{\bar{x}}^{\bar{v}}(\delta)) = F$;
- $\perp \vdash R_{\bar{x}}^{\bar{v}}(P) \in H \Rightarrow P_A(r_{\bar{x}}^{\bar{v}}(\delta)) \uparrow$.
- Formulas of the form $\vdash R_{\bar{x}}^{\bar{v}}(P)$ are called *primitive*.

For a predicate symbol $P \in Ps$ that does not occur in H , its value can be chosen in arbitrary way. Also, we should treat variables from U as unessential.

For atomic and primitive formulas the satisfiability statements follow from their definitions.

Now the proof goes on by induction on the formula structure.

Let us prove the theorem for conditions HRI_L , $HR\Box_R$, $HR\Box_U$, $H\neg_L$, $H\neg_R$, $H\neg_U$, $H\vee_L$, $H\vee_R$, $H\vee_U$, $H\Box_U$, $H\Box_L$.

Let $\vdash R_{z, \bar{x}}^{z, \bar{v}}(\Phi) \in H$. By HRI_L we have $\vdash R_{\bar{x}}^{\bar{v}}(\Phi) \in H$. By induction hypothesis $R_{\bar{x}}^{\bar{v}}(\Phi)_A(\delta) = T$, therefore $R_{z, \bar{x}}^{z, \bar{v}}(\Phi)_A(\delta) = T$.

Let $\neg \vdash R_{\bar{x}}^{\bar{v}}(\Box \Phi) \in H$. By $HR\Box_R$ we have $\neg \vdash R_{\bar{x}}^{\bar{v}}(\Phi) \in H$. By induction hypothesis $\Box R_{\bar{x}}^{\bar{v}}(\Phi)_A(\delta) = F$, therefore $R_{\bar{x}}^{\bar{v}}(\Box \Phi)_A(\delta) = F$.

Let $\perp \vdash R_{\bar{x}}^{\bar{v}}(\Box \Phi) \in H$. By $HR\Box_U$ we have $\perp \vdash R_{\bar{x}}^{\bar{v}}(\Phi) \in H$. By induction hypothesis $\Box R_{\bar{x}}^{\bar{v}}(\Phi)_A(\delta) \uparrow$, therefore $R_{\bar{x}}^{\bar{v}}(\Box \Phi)_A(\delta) \uparrow$.

Let $\vdash \neg \Phi \in H$. By $H\neg_L$ we have $\neg \vdash \Phi \in H$. By induction hypothesis $\Phi_A(\delta) = F$, therefore $\neg \Phi_A(\delta) = T$.

Let $\neg \vdash \neg \Phi \in H$. By $H\neg_R$ we have $\vdash \Phi \in H$. By induction hypothesis $\Phi_A(\delta) = T$, therefore $\neg \Phi_A(\delta) = F$.

Let $\vdash \Phi \vee \Psi \in H$. By $H\vee_L$ we have $\vdash \Phi \in H$ or $\vdash \Psi \in H$. By induction hypothesis $\Phi_A(\delta) = T$ and $\Psi_A(\delta) = T$, therefore $(\Phi \vee \Psi)_A(\delta) = T$.

Let $\neg \vdash \Phi \vee \Psi \in H$. By $H\vee_R$ we have $\neg \vdash \Phi \in H$ and $\neg \vdash \Psi \in H$. By induction hypothesis $\Phi_A(\delta) = F$ and $\Psi_A(\delta) = F$, therefore $(\Phi \vee \Psi)_A(\delta) = F$.

Let $\perp \vdash \neg \Phi \in H$. By $H\neg_U$ we have $\perp \vdash \Phi \in H$. By induction hypothesis $\Phi_A(\delta) \uparrow$, therefore $\neg \Phi_A(\delta) \uparrow$.

Let $\perp \vdash \Phi \vee \vartheta \in H$. By $H\vee_U$ $\perp \vdash \Phi \in H$ and $\perp \vdash \vartheta \in H$ or $\perp \vdash \Phi \in H$ and $\neg \vdash \vartheta \in H$ or $\neg \vdash \Phi \in H$ and $\perp \vdash \vartheta \in H$. By induction hypothesis $\Phi_A(\delta) \uparrow$ and $\vartheta_A(\delta) \uparrow$ or $\Phi_A(\delta) \uparrow$ and $\vartheta_A(\delta) = F$ or $\Phi_A(\delta) = F$ and $\vartheta_A(\delta) \uparrow$. Therefore $(\Phi \vee \vartheta)_A(\delta) \uparrow$.

Let $\perp \Box \Phi \in H$. By $H \Box \cup$ we have $\perp \Phi \in H$ or $\neg \Phi \in H$. By induction hypothesis $\Phi_A(\delta) = T$ or $\Phi_A(\delta) = F$, this gives $\Phi_A(\delta) \downarrow$, therefore $\Box \Phi_A(d) \uparrow$.

Let $\perp \Box \Phi \in H$. By $H \Box \perp$ we have $\perp \Phi \in H$. By induction hypothesis $\Phi_A(\delta) \uparrow$, therefore $\Box \Phi_A(d) = T$.

Theorem 7. Let \wp be unclosed path in a sequent tree for $\perp \Gamma \perp U \neg \Delta$ and H be the set of all formulas in \wp . Then H is a model set.

Proof. We should check that H satisfies all requirements that specify a model set. Details can be found in [12] but additionally undefinedness conditions should be taken into account.

The completeness theorem follows from Theorems 6 and 7.

Theorem 8 (completeness). Let $U/\Gamma \models_{IR} \Delta$ hold. Then sequent $\perp \Gamma \perp U \neg \Delta$ is derivable in C^{QCR} .

Proof. Assume that $U/\Gamma \models_{IR} \Delta$ and $\perp \Gamma \perp U \neg \Delta$ is not derivable. In this case a sequent tree for $\perp \Gamma \perp U \neg \Delta$ is not closed. Thus, an unclosed path \wp exists in this tree. Let H be the set of all formulas of this path. By Theorem 7, H is a model set. By theorem 6 this means that a counter-model for $\perp \Gamma \perp U \neg \Delta$ was constructed. But this contradicts to $U/\Gamma \models_{IR} \Delta$.

5. Conclusion

The efficiency of program verification heavily depends on program logics supporting corresponding verification methods. Traditional Floyd-Hoare logic and its variants are oriented on total pre- and post-conditions (total predicates) and do not support partial predicates. In this paper we have studied a new method for constructing sound program logics. This method is based on extending program logics with the composition of predicate complement. The method permits to construct a sound calculus for program logic but it makes the calculus more complicated because undefinedness conditions should be taking into consideration.

Also, introduction of partial predicates required extension of a base predicate logic to a logic of partial quasiary predicate. For this logic we have defined and investigated a special consequence relation called irrefutability consequence relation with undefinedness conditions. For a case of quantifier-free predicate logic of partial quasiary predicates (renominative level) we have constructed a calculus of sequent type and proved its soundness and completeness.

In the future we plan to construct a sequent calculus for predicate logic over hierarchical nominative data and prove its soundness and completeness. Also, we plan to develop a prototype of theorem prover oriented on such logics. Initial steps were made in [15].

References

- [1] Abramsky, S., Gabbay, D., Maibaum, T. (editors). Handbook of Logic in Computer Science, Oxford University Press, Vol. 1–5 (1993–2000)

- [2] Hoare, C.: An axiomatic basis for computer programming, *Commun. ACM* 12(10), 576–580 (1969)
- [3] Apt, K.: Ten years of Hoare’s logic: a survey - part I, *ACM Trans. Program. Lang. Syst.* 3(4), 431–483 (1981)
- [4] Kryvolap, A., Nikitchenko, M., Schreiner, W.: Extending Floyd-Hoare logic for partial pre- and postconditions. In: *Communications in Computer and Information Science*, vol. 412, pp. 355–378. Springer, Cham (2013)
- [5] Nikitchenko, M., Kryvolap, A.: Properties of inference systems for Floyd-Hoare logic with partial predicates, *Acta Electrotechnica et Informatica*, 13(4), 70–78 (2013)
- [6] Ivanov, I., Nikitchenko, M.: On the sequence rule for the Floyd-Hoare logic with partial pre- and post-conditions. In *Proceedings of the 14th International Conference on ICT*, vol. 2104 of *CEUR Workshop Proc.*, pp. 716–724 (2018)
- [7] Nikitchenko, M., Ivanov, I., Kornilowicz, A., Kryvolap, A.: Extended Floyd-Hoare logic over relational nominative data. In: Bassiliades, N., Ermolayev, V., Fill, H.G., Yakovyna, V., Mayr, H.C., Nikitchenko, M., Zholtkevych, G., Spivakovsky, A. (eds.) *Information and Communication Technologies in Education, Research, and Industrial Applications*. pp. 41–64. Springer International Publishing, Cham (2018)
- [8] Nikitchenko, N.S.: A composition-nominative approach to program semantics. Technical report, IT-TR 1998–020, Technical University of Denmark (1998)
- [9] Nikitchenko, M., Shkilniak, S.: *Mathematical logic and theory of algorithms*. Kyiv: VPC Kyivskiyi Universytet, in Ukrainian (2008)
- [10] Nikitchenko, M., Tymofieiev, V.: Satisfiability in Composition-Nominative Logics, *Open Computer Science (former Central European Journal of Computer Science)*, vol. 2, issue 3, pp. 194–213 (2012)
- [11] Nikitchenko, M., Ivanov, I., Skobelev, V.: Proving properties of programs on hierarchical nominative data, *Comput. Sci. J. Moldova*, 24(3(72)), 371–398 (2016)
- [12] Nikitchenko, M., Shkilniak, S.: Algebras and logics of partial quasiary predicates, *Algebra and Discrete Mathematics*, vol. 23. No 2, pp. 263–278 (2017)
- [13] Kornilowicz, A., Ivanov, I., Nikitchenko, M.: Kleene algebra of partial predicates. *Formalized Mathematics*, 26, 11–20 (2018)
- [14] Nikitchenko, M., Shkilniak, O. Shkilniak, S.: Pure first-order logics of quasiary predicates, *Problems in Programming*. No 2–3, pp. 73–86, in Ukrainian (2016)
- [15] Kornilowicz, A., Kryvolap, A., Nikitchenko, M., Ivanov, I.: Formalization of the nominative algorithmic algebra in Mizar. In: J. Świątek, L. Borzemski, Z. Wilimowska (eds.) *ISAT 2017, AISC*, vol. 656, pp. 176–186. Springer, Cham (2018)