UDC 510.64
DOI: https://doi.org/10.17721/1812-5409.2024/2.13

Stepan SHKILNIAK, DSc (Phys. & Math.), Prof.
ORCID ID: 0000-0003-8624-5778
e-mail: ss.sh@knu.ua
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine

VARIETIES OF PURE FIRST-ORDER LOGICS OF PARTIAL QUASIARY PREDICATES

Classical predicate logic typically lies at the basis of various logical systems successfully used in computer science and programming. However, classical logic has fundamental limitations that complicate its application. Therefore, the task of developing new, program-oriented logics becomes important; this highlights the relevance of the proposed research. Composition-nominative logics of partial quasiary predicates are the important class of such logics. The paper continues the research of pure first-order compositional-nominative logics (PCNL). New classes of PCNL are obtained by introducing partial predicates-indicators. Special predicates-indicators determine the presence of components with the corresponding subject name in the input data, which is necessary to consider for the quantifier elimination in the logics of non-monotone predicates. Two types of these predicates can be distinguished: total Ez, which determines the presence or absence of a component with a given name z, and partial, which only detects the presence of such a component. We propose partial predicates-indicators ↓z and the corresponding classes of PCNL. Depending on the presence and type of predicates-indicators, equality predicates (weak $=_{xy}$ or strong $\equiv_{xy}$), and the use of traditional or extended renomination, in this paper we specify a number of classes of PCNL. At the basic level, we have L^{Q} logics with traditional renominations and $L_{\perp}^{Q}$ logics with extended renominations. Extending these logics with the predicates-indicators ↓z yields new logics $L_{\downarrow}^{Q}$ and $L_{\downarrow\perp}^{Q}$, and their extension with the predicates Ez gives us the logics L_{I}^{Q} and $L_{\mathrm{I}\perp}^{Q}$. The further extension with the equality predicates $=_{xy}$ and $\equiv_{xy}$ results in the corresponding logics $L^{Q=}$, $L_{\perp}^{Q=}$ and $L^{Q\equiv}$, $L_{\perp}^{Q\equiv}$. The basic compositions of these logics are described, and their properties are presented. The languages of the introduced classes of L^{Q} are specified, and a number of logical consequence relations for these languages are defined. The characteristics of these relations are investigated, and the relationships between them are provided.

K e y w o r d s : ***logic, partial predicate, equality, logical consequence.***

AMS 2020 classification: 62H30, 34C60.

Introduction

The apparatus of mathematical logic is the foundation of modern information and software systems, as well as artificial intelligence systems (see, e. g., Abramsky, Gabbay, & Maibaum, 1993–2000; Bjorner, & Henson, 2008; Huth, & Ryan, 2004). Classical logic of predicates is typically used for this purpose. This logic is well studied (see, e. g., Barwise, 1977; Kleene, 2013; Shoenfield, 1967), has a rich history of applications, and it serves as the basis for constructing special logics aimed at solving specific problems (modal, temporal, epistemic, program logics etc.). However, classical logic has fundamental limitations that complicate its application. This logic is based on traditional mathematical structures of single-valued total finite-ary mappings, while the characteristic feature of programming and modeling is the use of partial, and sometimes many-valued (non-deterministic), mappings over complex data. Therefore, the problem of developing new logical formalisms more adapted to the needs of informatics and programming becomes relevant. This important topic is the focus of this work.

The composition-nominative approach, which is common to both programming and logic, can be the natural basis for the construction of program-oriented logics. Logics built on this foundation are called composition-nominative logics (CNL). These logics are based on classes of partial mappings defined on nominative data. Such mappings are called quasiary, hence we speak of CNL of partial quasiary predicates. Several classes of such logics are described, for instance, in (Nikitchenko, Shkilniak, O., & Shkilniak, S., 2016; Shkilniak, 2019; Shkilniak, O., & Shkilniak, S., 2023).

In this work we investigate pure first-order CNL (PCNL). Such logics will be called L^{Q} (logics of quantifier level). We will use the same name for PCNL of the base level with traditional renomination compositions. L^{Q} with compositions of extended renomination will be called $L_{\perp}^{Q}$. Logics with extended renominations are discussed, for instance, in (Shkilniak, O., & Shkilniak, S., 2023). For quantifier elimination in logics of non-monotone predicates, the presence of components with corresponding subject names in the input data must be considered. Therefore, to describe the quantifier elimination, we use special 0-ary compositions – predicates-indicators which determine such a presence. Two types of these predicates can be considered: total, which determines the *presence* or *absence* of a component with a given name, and partial, which only detects the *presence* of such a component. Total predicates-indicators have been investigated, for instance, in (Nikitchenko, Shkilniak, O., & Shkilniak, S., 2016; Shkilniak, 2019). In this paper, we propose partial predicates-indicators ↓*z*. The predicates-indicators allow further differentiation of PCNL varieties: L^{Q} and $L_{\perp}^{Q}$ with total predicates *Ez* will be called L_{I}^{Q} and $L_{\mathrm{I}\perp}^{Q}$; L^{Q} and $L_{\perp}^{Q}$ with partial predicates ↓*z* will be called $L_{\downarrow}^{Q}$ and $L_{\downarrow\perp}^{Q}$. Note that in work (Shkilniak, O., & Shkilniak, S., 2023) $L_{\mathrm{I}\perp}^{Q}$ logics are referred to as $L_{\perp}^{Q}$.

Special equality predicates allow for equalling and distinguishing the values of subject names. Two types of such predicates can be considered (Shkilniak, 2019): weak equality predicates $=_{xy}$ and strong equality predicates $\equiv_{xy}$. Equality predicates have a general logical status, so we treat them as special 0-ary compositions. We obtain four types of PCNL with equality: $L^{Q=}$, $L_{\perp}^{Q=}$ and $L^{Q\equiv}$, $L_{\perp}^{Q\equiv}$. The actual presence of predicates-indicators in these logics is important. In $L^{Q=}$ and $L_{\perp}^{Q=}$, there is no need to introduce additional predicates since $=_{xy}$ and ↓*x* are the *same* predicate. The predicates *Ez*, which operate in $L^{Q\equiv}$ and $L_{\perp}^{Q\equiv}$, are represented as $\exists y \equiv_{zy}$, making them *expressible* in $L^{Q\equiv}$ and $L_{\perp}^{Q\equiv}$. However, it is not practical to consider *Ez* as complex derivative predicates, especially since *Ez* predicates can be specified on the renominative level. Therefore, we explicitly define *Ez* as 0-ary compositions.

Thus, depending on the use of traditional or extended renomination, the presence and type of predicates-indicators (total or partial), and equality predicates (weak or strong), in this paper we distinguish corresponding classes of PCNL. The basic compositions of these logics are described, and their properties are presented. The languages of the introduced classes of PCNL are specified, and a number of logical consequence relations for these languages are defined. The characteristics of these relations are investigated, and the relationships between them are provided.

Concepts not defined here are interpreted in the sense of works (Nikitchenko, Shkilniak, O., & Shkilniak, S., 2016; Shkilniak, 2019; Shkilniak, O., & Shkilniak, S., 2023).

1. Composition predicate algebras of PCNL

The core semantic concept of CNL is the notion of a quasiary predicate. These are predicates defined over sets of named values, known as nominative sets.

We define *V-A*-nominative set (*V-A*-IM) as a single-valued function of the form $d: V \to A$, where V and A are sets of subject names (variables) and subject values (basic data) respectively. *V-A*-IM is represented as a set of pairs $[v_1 \mapsto a_1,...,v_n \mapsto a_n,...]$, where $v_i \in V$, $a_i \in A$, $\neq v_j$ v_i when $i \neq j$. A set of all *V-A*-IM will be denoted by ${}^VA$.

We introduce a parametric operation $||_{-X}$, where $X \subseteq V$, as follows: $d||_{-X} = \{v \mapsto a \in d \mid v \notin X\}$. A parametric operation of the extended renomination $\mathrm{r}^{[v_1 \mapsto x_1,...,v_n \mapsto x_n, u_1 \mapsto \perp,...,u_m \mapsto \perp]}: {}^VA \to {}^VA$ is specified as (Shkilniak, O., & Shkilniak, S., 2023):

$$\mathrm{r}^{[v_1 \mapsto x_1,...,v_n \mapsto x_n, u_1 \mapsto \perp,...,u_m \mapsto \perp]}(d) =$$
$$= [v_1 \mapsto d(x_1),...,v_n \mapsto d(x_n)] \cup d\,||_{-v_1,...,v_n,u_1,...,u_m}.$$

Here v_i, $x_i \in V$; the special symbol $\perp \notin V$ denotes the absence of the variable value.

Simpler notation for $\mathrm{r}^{[v_1 \mapsto x_1,...,v_n \mapsto x_n, u_1 \mapsto \perp,...,u_m \mapsto \perp]}$ can be used: either $\mathrm{r}^{v_1,...,v_n,u_1,...,u_m}_{x_1,...,x_n,\perp,...,\perp}$, or even more compact $\mathrm{r}^{\bar{v},\bar{u}}_{\bar{x},\perp}$. Traditional renomination operation $\mathrm{r}^{\bar{v}}_{\bar{x}}: {}^VA \to {}^VA$ (Nikitchenko, Shkilniak, O., & Shkilniak, S., 2016) is a special case of $\mathrm{r}^{\bar{v},\bar{u}}_{\bar{x},\perp}$.

V-A-quasiary predicate is specified by a partial, and generally speaking, many-valued function $Q: {}^VA \to \{T, F\}$, where $\{T, F\}$ is the set of truth values. In this paper, many-valued *V-A*-quasiary predicates are treated as relations between ${}^VA$ and $\{T, F\}$, thus they are called *R*-predicates.

Each *R*-predicate Q is uniquely defined by two sets:

- the truth domain (*T*-domain) $T(Q) = \{d \in {}^VA \mid T \in Q[d]\}$;
- the falsity domain (*F*-domain) $F(Q) = \{d \in {}^VA \mid F \in Q[d]\}$.

Here $Q[d]$ denotes the set of values, which *R*-predicate Q assumes on argument (data) d. Such $Q[d]$ can be one of $\varnothing$, $\{T\}$, $\{F\}$, $\{T, F\}$. For single-valued predicates, the usual notation $Q(d)$ will be used.

We distinguish (Nikitchenko, Shkilniak, O., & Shkilniak, S., 2016) a number of classes of *R*-predicates as follows.

V-A-quasiary *R*-predicate Q is:

- partial single-valued, or P-predicate, if $T(Q) \cap F(Q) = \varnothing$;
- total, or T-predicate, if $T(Q) \cup F(Q) = {}^VA$;
- irrefutable, or partially true, if $F(Q) = \varnothing$;
- satisfiable, if $T(Q) \neq \varnothing$;
- totally true (denoted T), if $F(Q) = \varnothing$ and $T(Q) = {}^VA$;
- totally false (denoted F), if $T(Q) = \varnothing$ and $F(Q) = {}^VA$;
- totally undefined (denoted $\perp$), if $T(Q) = F(Q) = \varnothing$;
- totally ambivalent (denoted Υ), if $T(P) = F(P) = {}^VA$;
- total single-valued, or TS-predicate, if TS-predicate, if $T(Q) \cap F(Q) = \varnothing$ and $T(Q) \cup F(Q) = {}^VA$;
- monotone, if $d1 \subseteq d2 \Rightarrow Q[d1] \subseteq Q[d2]$;
- antitone, if $d_1 \subseteq d_2 \Rightarrow Q[d_1] \supseteq Q[d_2]$.

For *P*-predicates, monotonicity becomes equitonicity: Q is equitone, if $(Q(d)\downarrow$ and $d \subseteq d') \Rightarrow Q(d')\downarrow = Q(d)$.

We will denote classes of *V-A*-quasiary *R*-predicates, *P*-predicates, *T*-predicates, *TS*-predicates, monotone *R*-predicates, antitone *R*-predicates, equitone *P*-predicates and antitone *T*-predicates as PrR^{V-A}, PrP^{V-A}, PrT^{V-A}, $PrTS^{V-A}$, $PrRM^{V-A}$, $PrRA^{V-A}$, $PrPE^{V-A}$ and $PrTA^{V-A}$ respectively.

Predicates of classical logic are total single-valued and monotone (equitone). Logics of equitone *P*-predicates are a natural extension of classical logic and preserve its main properties. However, all *TS*-predicates, except constant T and F, are non-monotone. Therefore, the class $PrTS^{V-A}$ is degenerate in this sense.

We obtain 4 constant *R*-predicates: T, F, $\perp$, Υ. They are monotone and antitone.

A name $x \in V$ is *unessential* for $Q \in PrR^{V-A}$, if for any $d_1, d_2 \in {}^VA$ we have: $d_1||_{-x} = d_2||_{-x} \Rightarrow Q(d_1) = Q(d_2)$.

R-predicate $\tilde{Q}$ is *dual* to *R*-predicate Q, if $T(\tilde{Q}) = \overline{F(Q)}$ and $F(\tilde{Q}) = \overline{T(Q)}$.

The dualization mapping $\delta: PrR^{V-A} \to PrR^{V-A}$ is defined as $\delta(Q) = \tilde{Q}$.

Statement 1. $\delta(\mathrm{T}) = \mathrm{T}$, $\delta(\mathrm{F}) = \mathrm{F}$, $\delta(\perp) = \Upsilon$, $\delta(\Upsilon) = \perp$; $\delta(PrR^{V-A}) = PrR^{V-A}$, $\delta(PrTS^{V-A}) = PrTS^{V-A}$;
$\delta(PrP^{V-A}) = PrT^{V-A}$, $\delta(PrT^{V-A}) = PrP^{V-A}$;
$\delta(PrPE^{V-A}) = PrTA^{V-A}$, $\delta(PrTA^{V-A}) = PrPE^{V-A}$;
$\delta(PrRM^{V-A}) = PrRA^{V-A}$, $\delta(PrRA^{V-A}) = PrRM^{V-A}$.

We describe the basic compositions of PCNL. Logical connectives $\neg$ and $\vee$ we define as:

$$T(\neg P) = F(P);\quad F(\neg P) = T(P);$$
$$T(P \vee Q) = T(P) \cup T(Q);\quad F(P \vee Q) = F(P) \cap F(Q).$$

We specify the compositions $\mathrm{R}^{\bar{v}}_{\bar{x}}: PrR^{V-A} \to PrR^{V-A}$ and $\mathrm{R}^{\bar{v},\bar{u}}_{\bar{x},\perp}: PrR^{V-A} \to PrR^{V-A}$ as follows:

$$\mathrm{R}^{\bar{v},\bar{u}}_{\bar{x},\perp}(Q)[d] = Q\,[\mathrm{r}^{\bar{v},\bar{u}}_{\bar{x},\perp}(d)];\quad \mathrm{R}^{\bar{v}}_{\bar{x}}(Q)[d] = Q\,[\mathrm{r}^{\bar{v}}_{\bar{x}}(d)].$$

We define the composition of quantification (quantifier) $\exists x : PrR^{V-A} \to PrR^{V-A}$ as follows:

$$T(\exists xQ) = \bigcup_{a\in A} \{d \mid d\,\|_{-x} \cup x \mapsto a \in T(Q)\};$$

$$F(\exists xQ) = \bigcap_{a\in A} \{d \mid d\,\|_{-x} \cup x \mapsto a \in F(Q)\}.$$

The derivative compositions $\&$, $\to$, $\forall x$ are specified traditionally:

$$P\&Q = \neg(\neg P\vee\neg Q),\ P\to Q = \neg P\vee Q,\ \ \forall xQ = \neg\exists x\neg P.$$

Special parametric predicates-indicators detect the presence of a component with a given subject name in the input data. There are two types of such predicates. Total predicates Ez (Nikitchenko, Shkilniak, O., & Shkilniak, S., 2016) determine the *presence* or *absence* of a component with the name $z\in V$ in the input data. They are defined as follows:

$$T(Ez) = \{d \mid d(z)\downarrow\};\quad F(Ez) = \{d \mid d(z)\uparrow\}.$$

In this paper we introduce partial predicates-indicators $\downarrow z$, which only detect the *presence* of a component with the name $z\in V$ in the input data. They can be defined as follows:

$$T(\downarrow z) = \{d \mid d(z)\downarrow\};\quad F(\downarrow z) = \varnothing.$$

Therefore, $T(Ez) = T(\downarrow z)$; however, $F(\downarrow z) = \varnothing$.

Ez predicates are non-monotone, $\downarrow z$ predicates are equitone. Each $x\in V$ such that $x\neq z$ is unessential for both Ez and $\downarrow z$.

Equality predicates parameterized by names allow for equaling and distinguishing the subject names' values. Two types of such predicates are specified (Shkilniak, 2019): weak (up to definability) equality predicates $=_{\{x,y\}}$ and strong (strict) equality predicates $\equiv_{\{x,y\}}$. They are defined as follows:

$T(=_{\{x,y\}}) = \{d \mid d(x)\downarrow,\ d(y)\downarrow \text{ and } d(x) = d(y)\}$,
$F(=_{\{x,y\}}) = \{d \mid d(x)\downarrow,\ d(y)\downarrow \text{ and } d(x) \neq d(y)\}$;
$T(\equiv_{\{x,y\}}) = \{d \mid d(x)\downarrow,\ d(y)\downarrow \text{ and } d(x) = d(y)\} \cup \{d \mid d(x)\uparrow \text{ and } d(y)\uparrow\}$,
$F(\equiv_{\{x,y\}}) = \{d \mid d(x)\downarrow,\ d(y)\downarrow,\ d(x) \neq d(y)\} \cup \{d \mid d(x)\downarrow,\ d(y)\uparrow \text{ or } d(x)\uparrow,\ d(y)\downarrow\}$.

Predicates $\equiv_{\{x,y\}}$ are total single-valued and non-monotone; predicates $=_{\{x,y\}}$ are partial and equitone. In case x and y match, predicates $=_{\{x,y\}}$ and $\equiv_{\{x,y\}}$ become $=_{\{x\}}$ and $\equiv_{\{x\}}$. Further on, predicates $=_{\{x,y\}}$, $=_{\{x\}}$ and $\equiv_{\{x,y\}}$, $\equiv_{\{x\}}$ will be denoted more compactly by $=_{xy}$, $=_{xx}$ and $\equiv_{xy}$, $\equiv_{xx}$. Thus, the pairs $=_{xy}$ and $=_{yx}$, $\equiv_{xy}$ and $\equiv_{yx}$ represent the same predicates. Therefore, the *symmetric* property of equality lies within notation.

Statement 2. *For equality properties we obtain the reflexivity and transitivity properties:*

- *Rf=) each predicate $=_{xx}$ is irrefutable;*
- *Rf≡) each predicate $\equiv_{xx}$ is the constant* T, *i.e.* $\equiv_{xx} = $ T*;*
- *Tr=) for all $d\in {}^VA$ we have: $=_{xy}(d) = T$ and $=_{yz}(d) = T \Rightarrow =_{xz}(d) = T$;*
- *Tr≡) for all $d\in {}^VA$ we have: $\equiv_{xy}(d) = T$ and $\equiv_{yz}(d) = T \Rightarrow \equiv_{xz}(d) = T$.*

Properties Tr= and Tr≡ imply: predicates $=_{xy} \& =_{yz} \to =_{xz}$ are irrefutable; $\equiv_{xy} \& \equiv_{yz} \to \equiv_{xz} = $ T.

Statement 3. *$=_{xx}$ and $\downarrow x$ represent the same predicate: $=_{xx} = \downarrow x$.*

This means that in $L^{Q=}$ and $L_{\perp}{}^{Q=}$ predicates-indicators $\downarrow x$ are *already present* in the form of $=_{xx}$ predicates!

Therefore, depending on the use of traditional or extended renomination, the presence and type of predicates-indicators (total Ez or partial $\downarrow z$), and equality predicates (weak $=_{xy}$ or strong $\equiv_{xy}$), we distinguish the following classes of PCNL:

- L^Q with the set of basic compositions $C_Q = \{\neg, \vee, R^{\bar v}_{\bar x}, \exists x\}$;
- $L_{\perp}{}^Q$ with the set of basic compositions $C_{\perp Q} = \{\neg, \vee, R^{\bar v,\bar u}_{\bar x,\perp}, \exists x\}$;
- $L_{\downarrow}{}^Q$ with the set of basic compositions $C_{\downarrow Q} = \{\neg, \vee, R^{\bar v}_{\bar x}, \exists x, \downarrow z\}$;
- $L_{\downarrow\perp}{}^Q$ with the set of basic compositions $C_{\downarrow\perp Q} = \{\neg, \vee, R^{\bar v,\bar u}_{\bar x,\perp}, \exists x, \downarrow z\}$;
- $L_{\mathrm{I}}{}^Q$ with the set of basic compositions $C_{\mathrm{I}Q} = \{\neg, \vee, R^{\bar v}_{\bar x}, \exists x, Ez\}$;
- $L_{\mathrm{I}\perp}{}^Q$ with the set of basic compositions $C_{\mathrm{I}\perp Q} = \{\neg, \vee, R^{\bar v,\bar u}_{\bar x,\perp}, \exists x, Ez\}$;
- $L^{Q=}$ with the set of basic compositions $C_{Q=} = \{\neg, \vee, R^{\bar v}_{\bar x}, \exists x, =_{xy}\}$;
- $L_{\perp}{}^{Q=}$ with the set of basic compositions $C_{\perp Q=} = \{\neg, \vee, R^{\bar v,\bar u}_{\bar x,\perp}, \exists x, =_{xy}\}$;
- $L^{Q\equiv}$ with the set of basic compositions $C_{Q\equiv} = \{\neg, \vee, R^{\bar v}_{\bar x}, \exists x, Ez, \equiv_{xy}\}$;
- $L_{\perp}{}^{Q\equiv}$ with the set of basic compositions $C_{\perp Q\equiv} = \{\neg, \vee, R^{\bar v,\bar u}_{\bar x,\perp}, \exists x, Ez, \equiv_{xy}\}$.

The semantic basis for PCNL of a given class is formed by composition predicate systems $({}^VA, Pr^{V-A}, C_B)$, where C_B is a set of basic compositions, Pr^{V-A} is a class of quasiary predicates. For the introduced PCNL classes, composition systems of R-predicates are tuples $({}^VA, PrR^{V-A}, C_{*Q*})$. Each such system defines a data algebra (structure, algebraic system) $({}^VA, Pr^{V-A})$ and a composition predicate algebra (PrR^{V-A}, C_{*Q*}).

Thus, we obtained the first-order composition algebras of R-predicates:

- $A^Q = (PrR^{V-A}, C_Q)$, $A^{\perp Q} = (PrR^{V-A}, C_{\perp Q})$;
- $A^{\downarrow Q} = (PrR^{V-A}, C_{\downarrow Q})$, $A^{\downarrow\perp Q} = (PrR^{V-A}, C_{\downarrow\perp Q})$;

- $A^{IQ} = (PrR^{V\text{–}A}, C_{IQ})$, $A^{I\perp Q} = (PrR^{V\text{–}A}, C_{I\perp Q})$;
- $A^{Q=} = (PrR^{V\text{–}A}, C_{Q=})$, $A^{\perp Q=} = (PrR^{V\text{–}A}, C_{\perp Q=})$;
- $A^{Q\equiv} = (PrR^{V\text{–}A}, C_{Q\equiv})$, $A^{\perp Q\equiv} = (PrR^{V\text{–}A}, C_{\perp Q\equiv})$.

All the defined above classes of predicates are closed under the compositions C_Q and $C_{\perp Q}$.

The classes $PrP^{V\text{–}A}$, $PrRM^{V\text{–}A}$, $PrPE^{V\text{–}A}$ are closed under the compositions $C_{\downarrow Q}$, $C_{\downarrow\perp Q}$ and $C_{Q=}$, $C_{\perp Q=}$. The classes $PrRM^{V\text{–}A}$, $PrRA^{V\text{–}A}$, $PrPE^{V\text{–}A}$, $PrTA^{V\text{–}A}$ are closed under the compositions C_{IQ}, $C_{I\perp Q}$ and $C_{Q\equiv}$, $C_{\perp Q\equiv}$. This means that a number of corresponding subalgebras of the specified predicate algebras can be distinguished.

In the $A^{\perp Q\equiv}$ algebra we specify the subalgebras of *P*-predicates $A^{\perp QP\equiv}$, *T*-predicates $A^{\perp QT\equiv}$, *TS*-predicates $A^{\perp QT\equiv S}$; the subalgebras of *P*-, *T*-, *TS*-predicates can be specified in $A^{Q\equiv}$, $A^{I\perp Q}$, A^{IQ} in the same manner.

Let δ be a dualization mapping. We call algebras (Pr_1, *C*) and (Pr_2, *C*) *dual*, if $\delta(Pr_1) = Pr_2$ and $\delta(Pr_2) = Pr_1$.

In logics of types $L_{\downarrow}{}^{Q}$ and $L^{Q=}$, the basic compositions include the partial single-valued predicates $\downarrow z$ and $=_{xy}$. The dualization mapping transforms them into total many-valued predicates, which are not among 0-ary compositions in these logics. Therefore, the concept of dual predicate algebras does not apply to $L_{\downarrow}{}^{Q}$, $L_{\downarrow\perp}{}^{Q}$, $L^{Q=}$, $L_{\perp}{}^{Q=}$.

In the logics $L_{\downarrow}{}^{Q}$, $L_{\downarrow\perp}{}^{Q}$, $L^{Q=}$, $L_{\perp}{}^{Q=}$ we specify subalgebras of *P*-predicates $A^{\downarrow QP}$, $A^{\downarrow\perp QP}$, $A^{QP=}$, $A^{\perp QP=}$, and subalgebras of monotone predicates $A^{\downarrow QRM}$, $A^{\downarrow QPE}$, $A^{\downarrow\perp QRM}$, $A^{\downarrow\perp QPE}$, $A^{QRM=}$, $A^{QPE=}$, $A^{\perp QRM=}$, $A^{\perp QPE=}$.

Given the duality between $PrP^{V\text{–}A}$ and $PrT^{V\text{–}A}$ and the degeneracy of $PrTS^{V\text{–}A}$, we will focus on the logics of *R*-predicates and *P*-predicates.

2. Properties of compositions in PCNL

The properties of propositional compositions and quantification compositions, not related to renominations and equality predicates, are generally similar to the properties of the corresponding classical logical connectives and quantifiers of classical logic.

The properties of renomination compositions $R_{\perp}I$, $R_{\perp}U$, R↑, $R_{\perp}R$, $R_{\perp}\exists s$, $R_{\perp}\exists$ are presented (Shkilniak O., & Shkilniak, S., 2023). The corresponding properties of traditional renominations R, RI, RU, R¬, R∨, RR, R∃s, R∃ are described in (Nikitchenko, Shkilniak, O., & Shkilniak, S., 2016). These properties hold for all the specified classes of R-predicates.

Statement 4. *We have the following renomination properties for the Ez predicates:*

$R_{\perp E}$) $R^{\bar{v},\bar{u}}_{\bar{x},\perp}(Ez) = Ez$, *if* $z \notin \{\bar{v},\bar{u}\}$;

$R_{\perp Ev}$) $R^{\bar{v},\bar{u},z}_{\bar{x},\perp,y}(Ez) = Ey$;

$R_{\perp EF}$) $R^{\bar{v},\bar{u},z}_{\bar{x},\perp,\perp}(Ez) = F$; *in particular,* $R^{z}_{\perp}(Ez) = F$.

The properties of traditional renomination R_E and R_{Ev} can be specified in the similar manner.

Statement 5. *We have the following quantification properties for the Ez predicates:*

- $\exists x\, Ex = \neg\exists x \neg Ex = \forall x\, Ex = T$ *and* $\exists x \neg Ex = \neg\exists x\, Ex = \forall x \neg Ex = F$;
- $\exists y\, Ex = \forall y\, Ex = Ex$ *and* $\exists y \neg Ex = \forall y \neg Ex = \neg Ex$ *provided that* $x \neq y$.

Statement 6. *We have the following properties for the predicates-indicators ↓z in the logics* $L_{\downarrow\perp}{}^{Q}$ *and* $L_{\downarrow}{}^{Q}$:

- $\neg{\downarrow}z \vee {\downarrow}z = {\downarrow}z$; $\neg{\downarrow}z \,\&\, {\downarrow}z = \neg{\downarrow}z$; ${\downarrow}z \to {\downarrow}z = \neg{\downarrow}z \vee {\downarrow}z = {\downarrow}z$;
- $\exists x {\downarrow}x = \forall x {\downarrow}x = T$ *and* $\exists x \neg{\downarrow}x = \neg\exists x {\downarrow}x = \forall x \neg{\downarrow}x = F$;
- $\exists y {\downarrow}x = \forall y {\downarrow}x = {\downarrow}x$ *and* $\exists y \neg{\downarrow}x = \forall y \neg{\downarrow}x = \neg{\downarrow}x$ *provided that* $x \neq y$.

Statement 7. *We have the following renomination properties for the predicates-indicators ↓z*:

$R_{\perp\downarrow t}$) $R^{\bar{v},\bar{u}}_{\bar{x},\perp}({\downarrow}z) = {\downarrow}z$, if $z \notin \{\bar{v},\bar{u}\}$;

$R_{\perp\downarrow v}$) $R^{\bar{v},\bar{u},z}_{\bar{x},\perp,y}({\downarrow}z) = {\downarrow}y$;

$R_{\perp\downarrow\perp}$) $R^{\bar{v},\bar{u},z}_{\bar{x},\perp,\perp}({\downarrow}z) = \perp$; *in particular,* $R^{z}_{\perp}({\downarrow}z) = \perp$.

The properties of traditional renomination $R_{\downarrow t}$ and $R_{\downarrow v}$ are described in the similar manner.

Statement 8. *The relations between the predicates ↓z, Ez and ⊥:* ${\downarrow}z = Ez \vee \perp$ *and* $\neg{\downarrow}z = \neg Ez \,\&\, \perp$.

Theorem 1. *1)* $\equiv_{xy} = (=_{xy} \& Ex \& Ey) \vee (\neg Ex \& \neg Ey)$;

2) $=_{xy} = (\equiv_{xy} \& x{\downarrow} \& {\downarrow}y) \vee \neg{\downarrow}x \vee \neg{\downarrow}y$.

Statement 9. *We have the following extended renomination properties for predicates* $\equiv_{xy}$ *and* $=_{xy}$:

$R_{\perp\equiv xx}$) $R^{\bar{v},\bar{u}}_{\bar{w},\perp}(\equiv_{xx}) = T$;

$R_{\perp\equiv\perp}$) $R^{\bar{v},\bar{u},x,y}_{\bar{w},\perp,\perp,\perp}(\equiv_{xy}) = T$;

$R_{\perp\equiv 2}$) $R^{\bar{v},\bar{u},x,y}_{\bar{w},\perp,z,s}(\equiv_{xy}) = \equiv_{zs}$ *and* $R^{\bar{v},\bar{u},x}_{\bar{w},\perp,z}(\equiv_{xx}) = \equiv_{zz}$;

$R_{\perp\equiv 1}$) $R^{\bar{v},\bar{u},x}_{\bar{w},\perp,z}(\equiv_{xy}) = \equiv_{zy}$, *if* $y \notin \{\bar{u},\bar{v}\}$ *and* $x \neq y$;

$R_{\perp\equiv 0}$) $R^{\bar{v},\bar{u}}_{\bar{w},\perp}(\equiv_{xy}) = \equiv_{xy}$, *if* $x, y \notin \{\bar{u},\bar{v}\}$ *and* $x \neq y$;

$R_{\perp\equiv 1E}$) $R^{\bar{v},\bar{u},x}_{\bar{w},\perp,\perp}(\equiv_{xy}) = \neg Ey$, *if* $y \notin \{\bar{u},\bar{v}\}$ *and* $x \neq y$;

$R_{\perp\equiv 2E}$) $R^{\bar{v},\bar{u},\,x,y}_{\bar{w},\perp,z,\perp}(\equiv_{xy}) = \neg Ez$;

$R_{\perp=\perp}$) $R^{\bar{v},\bar{u},\,x}_{\bar{w},\perp,\perp}(=_{xy}) = \perp$; *in particular,* $R^{x}_{\perp}(=_{xy}) = \perp$;

$R_{\perp=2}$) $R^{\bar{v}}_{\bar{x}}, R^{\bar{v},\bar{u},x,y}_{\bar{w},\perp,z,s}(=_{xy}) = \;=_{zs}$ *and* $R^{\bar{v},\bar{u},x}_{\bar{w},\perp,z}(=_{xx}) = \;=_{zz}$;

$R_{\perp=1}$) $R^{\bar{v},\bar{u},x}_{\bar{w},\perp,z}(=_{xy}) = \;=_{zy}$, *if* $y \notin \{\bar{u},\bar{v}\}$ *and* $x \neq y$;

$R_{\perp=0}$) $R^{\bar{v},\bar{u}}_{\bar{w},\perp}(=_{xy}) = \;=_{xy}$, *if* $x, y \notin \{\bar{u},\bar{v}\}$.

The corresponding properties of traditional renomination: $R\equiv_{xx}$, $R\equiv_2$, $R\equiv_1$, $R\equiv_0$ and $R{=}_2$, $R{=}_1$, $R{=}_0$.

Due to the equivalence of the predicates $\downarrow z$ and $=_{zz}$, the properties $R_{\perp\downarrow t}$, $R_{\perp\downarrow v}$, $R_{\perp\downarrow\perp}$ and $R_{\downarrow t}$, $R_{\downarrow v}$ can be rewritten as the properties $R_{\perp}{=}_{zz0}$, $R_{\perp}{=}_{zz}$, $R_{\perp}{=}_{zz\perp}$ and $R{=}_{zz0}$, $R{=}_{zz}$ for the predicates $=_{zz}$. The quantification properties of the predicates $\downarrow z$ are thus rewritten as the quantification properties of the predicates $=_{zz}$.

Statement 10. *We have the following quantification properties for the* $\equiv_{xx}$ *predicates:*

$$\exists x \equiv_{xx} = \forall x \equiv_{xx} = T;\; \exists x \neg\equiv_{xx} = \neg\exists x \equiv_{xx} = \forall x \neg\equiv_{xx} = F;\; \exists y \equiv_{xx} = \forall y \equiv_{xx} = \equiv_{xx} = T.$$

Statement 11. *Given that* $x \neq y$, *we have* $\exists y \equiv_{xy} = Ex$.

Hence, the predicates Ex can be expressed in $L^{Q\equiv}$ and $L_{\perp}^{Q\equiv}$, making them derivative in $L^{Q\equiv}$ and $L_{\perp}^{Q\equiv}$. However, quantifier elimination with the use of predicates-indicators $\exists y \equiv_{xy}$ is complicated, so in this case it is not practical to treat Ex as derivative predicates.

Statement 12. *Given that* $x \neq y$, *we have* $\exists x =_{xy} \;=\; =_{yy} = \downarrow y = Ey \vee \perp$.

Statement 13. *The substitution of equals for* $\equiv_{xy}$ *and* $=_{xy}$. *For every* $P \in Pr^{A}$ *and* $d \in {}^{V}A$ *we have:*

$\equiv R_{\perp}r$) $\equiv_{xy}(d) = T \Rightarrow R^{\bar{v},\bar{u},z}_{\bar{w},\perp,x}(P)(d) = R^{\bar{v},\bar{u},z}_{\bar{w},\perp,y}(P)(d)$;

$=R_{\perp}r$) $=_{xy}(d) = T \Rightarrow R^{\bar{v},\bar{u},z}_{\bar{w},\perp,x}(P)(d) = R^{\bar{v},\bar{u},z}_{\bar{w},\perp,y}(P)(d)$.

For traditional renomination we can obtain the similar properties $\equiv$Rr and =Rr.

For quantifier elimination, predicates-indicators are used. Thus, for the logics $L_{\mathrm{I}\perp}^{Q}$ and $L_{\perp}^{Q\equiv}$ we have:

Theorem 2. $T(R^{\bar{u},\bar{w},x}_{\bar{v},\perp,y}(P)) \cap T(Ey) \subseteq T(R^{\bar{u},\bar{w}}_{\bar{v},\perp}(\exists xP))$ *and* $F(R^{\bar{u},\bar{w}}_{\bar{v},\perp}(\exists xP)) \cap T(Ey) \subseteq F(R^{\bar{u},\bar{w},x}_{\bar{v},\perp,y}(P))$.

For the logics L_{I}^{Q} and $L^{Q\equiv}$, the statement of Theorem 2 can be formulated for traditional renominations.

We have $T(=_{zz}) = T(\downarrow z) = T(Ez)$, therefore for $L_{\downarrow\perp}^{Q}$ we obtain

Theorem 3. $T(R^{\bar{u},\bar{w},x}_{\bar{v},\perp,y}(P)) \cap T(\downarrow y) \subseteq T(R^{\bar{u},\bar{w}}_{\bar{v},\perp}(\exists xP))$ *and* $F(R^{\bar{u},\bar{w}}_{\bar{v},\perp}(\exists xP)) \cap T(\downarrow y) \subseteq F(R^{\bar{u},\bar{w},x}_{\bar{v},\perp,y}(P))$.

For $L_{\downarrow}^{Q}$, the latter statement is formulated for traditional renominations. For $L_{\perp}^{Q=}$, the statement of Theorem 3 will be rewritten changing $\downarrow y$ to $=_{yy}$; for $L^{Q=}$, traditional renominations will be additionally used.

3. Languages of PCNL

Depending on the use of traditional or extended renominations, the presence and type of predicates-indicators, and the presence and type of equality predicates, we have distinguished 10 varieties of PCNL and 10 corresponding classes of composition algebras of *R*-predicates. Terms of such composition algebras can be interpreted as formulas of the language of the corresponding PCNL class. In this context, the languages $L^{Q\equiv}$, $L^{Q=}$, L_{I}^{Q}, $L_{\downarrow}^{Q}$, L^{Q} are special cases of the languages $L_{\perp}^{Q\equiv}$, $L_{\perp}^{Q=}$, $L_{\mathrm{I}\perp}^{Q}$, $L_{\downarrow\perp}^{Q}$, $L_{\perp}^{Q}$, respectively, since traditional renominations are a special case of extended renominations. The languages $L_{\mathrm{I}\perp}^{Q}$ and L_{I}^{Q} are special cases of the languages $L_{\perp}^{Q\equiv}$ and $L^{Q\equiv}$, as they differ only by the absence of the $\equiv_{xy}$ predicates. The languages $L_{\downarrow\perp}^{Q}$ and $L_{\downarrow}^{Q}$ are essentially special cases of the languages $L_{\perp}^{Q=}$ and $L^{Q=}$, since the predicates $=_{zz}$ are actually predicates-indicators $\downarrow z$.

Thus, we can consider 3 groups of languages:

1) languages $L_{\perp}^{Q\equiv}$, $L^{Q\equiv}$, $L_{\mathrm{I}\perp}^{Q}$, L_{I}^{Q} with predicates Ez;
2) languages $L_{\perp}^{Q=}$, $L^{Q=}$, $L_{\downarrow\perp}^{Q}$, $L_{\downarrow}^{Q}$ with predicates $\downarrow z$;
3) languages $L_{\perp}^{Q}$, L^{Q} without explicitly defined predicates-indicators.

The presence the $\downarrow z$ predicates, asymmetrical with respect to their truth and falsity domains, introduces specific features into the semantic properties of the second group's languages.

Language of $L_{\perp}^{Q\equiv}$. The alphabet of the language consists of a set of names (variables) V, a set of predicate symbols Ps, and the set of symbols of basic compositions $Cs = \{\neg, \vee, R^{\bar{v},\bar{u}}_{\bar{x},\perp}, \exists x, Ez, \equiv_{xy}\}$.

Let us define inductively the set Fr of formulas of the language:

Fa) $Ps \subseteq Fr$;
FI) $\{Ex \mid x \in V\} \subseteq Fr$;
F≡) $\{\equiv_{xy} \mid x, y \in V\} \subseteq Fr$;
Fp) $\Phi, \Psi \in Fr \Rightarrow \neg\Phi \in Fr$ and $\vee\Phi\Psi \in Fr$;
FR) $\Phi \in Fr \Rightarrow R^{\bar{v},\bar{u}}_{\bar{x},\perp}\Phi \in Fr$;
F∃) $\Phi \in Fr \Rightarrow \exists x\Phi \in Fr$.

Formulas of the form $p\in Ps$, Ex, $\equiv_{xy}$ are called atomic.

To specify the sets of guaranteed to be unessential names for formulas of the language $L_{\perp}^{Q\equiv}$, we determine a set $V_T\subseteq V$ of names, unessential for all $p\in Ps$, and a function $\nu : Fr\to 2^V$. For all Ex and $\equiv_{xy}$, we set $\nu(Ex) = V\backslash\{x\}$ and $\nu(\equiv_{xy}) = V\backslash\{x, y\}$, for every $p\in Ps$ we define $\nu(p)$ considering $V_T\subseteq\nu(p)$.

Then, ν is described as:

$$\nu(\neg\Phi) = \nu(\Phi);\ \ \nu(\vee\Phi\Psi) = \nu(\Phi)\cap\nu(\Psi);\ \ \nu(\exists x\Phi) = \nu(\Phi)\cup\{x\};$$

$$\nu(R^{v_1,\dots,v_n,u_1,\dots,u_m}_{x_1,\dots,x_n,\perp,\dots,\perp}\Phi) = (\nu(\Phi)\cup\{v_1,\dots,v_n, u_1\dots,u_m\})\setminus\{x_i \mid v_i\notin\nu(\Phi),\ i\in\{1,\dots, n\}\}.$$

Let the tuple $\Sigma = (V, V_T, Cs, Ps)$ be called an extended signature of the language $L_{\perp}^{Q\equiv}$.

$\Phi\in Fr$ is called a *CF-formula*, if Φ does not contain compositions-constants symbols (Ex and $\equiv_{xy}$ for $L_{\perp}^{Q\equiv}$).

The language $L_{\perp}^{Q\equiv}$ is interpreted on composition systems CS = $({}^VA, PrR^{V\text{–}A}, C_{\perp Q\equiv})$. Symbols of basic compositions are interpreted as the corresponding compositions, Ex symbols as the corresponding predicates-indicators, and $\equiv_{xy}$ symbols as the corresponding equality predicates. To specify the basic predicates, we set the total single-valued I: $Ps\to PrR^{V\text{–}A}$ and extend it to the interpretation mapping for formulas I: $Ps\to PrR^{V\text{–}A}$:

$$I(\neg\Phi) = \neg(I(\Phi));\ \ I(\vee\Phi\Psi) = \vee(I(\Phi), I(\Psi));\ \ I(R^{\bar v,\bar u}_{\bar x,\perp}\Phi) = \mathrm{R}^{\bar v,\bar u}_{\bar x,\perp}(I(\Phi));\ I(\exists x\Phi) = \exists x(I(\Phi)).$$

Algebras with added signature $\boldsymbol{A} = (({}^VA, PrR^{V\text{–}A}), \Sigma, I)$ (the simpler notation $\boldsymbol{A} = (A, I)$) will be called interpretations of the language $L_{\perp}^{Q\equiv}$. Such algebras define the composition systems CS = $({}^VA, PrR^{V\text{–}A}, C_{\perp Q\equiv})$.

The specification of subalgebras of *P*-predicates $A^{\perp QP\equiv}$, *T*-predicates $A^{\perp QT\equiv}$ and *TS*-predicates $A^{\perp QTS\equiv}$ in the algebra of *R*-predicates defines a general class of *R*-interpretations and subclasses of *P*-interpretations, *T*- interpretations and *TS*-interpretations. These classes of interpretations are referred to as *semantics*.

Thus, in $L_{\perp}^{Q\equiv}$ we specified *R*-, *P*-, *T*-, and *TS*-semantics, which will be denoted by $_\perp\boldsymbol{R}^{\equiv}$, $_\perp\boldsymbol{P}^{\equiv}$, $_\perp\boldsymbol{T}^{\equiv}$, and $_\perp\boldsymbol{TS}^{\equiv}$. The $_\perp\boldsymbol{P}^{\equiv}$ and $_\perp\boldsymbol{T}^{\equiv}$ semantics form a dual pair, while the $_\perp\boldsymbol{R}^{\equiv}$ and $_\perp\boldsymbol{TS}^{\equiv}$ semantics are autodual.

Language of $L^{Q\equiv}$ is defined similarly to the language of $L_{\perp}^{Q\equiv}$, except instead of the symbols $R^{\bar v,\bar u}_{\bar x,\perp}$ we use $R^{\bar v}_{\bar x}$. The set of symbols of basic compositions is $\{\neg, \vee, R^{\bar v}_{\bar x}, \exists x, Ex, \equiv_{xy}\}$. The language $L^{Q\equiv}$ is interpreted in the same way as the language $L_{\perp}^{Q\equiv}$, but instead of $I(R^{\bar v,\bar u}_{\bar x,\perp}\Phi) = \mathrm{R}^{\bar v,\bar u}_{\bar x,\perp}(I(\Phi))$, we define $I(R^{\bar v}_{\bar x}\Phi) = \mathrm{R}^{\bar v}_{\bar x}(I(\Phi))$.

In $L^{Q\equiv}$ we specify *R*-, *P*-, *T*-, and *TS*-semantics, denoted by $\boldsymbol{R}^{\equiv}$, $\boldsymbol{P}^{\equiv}$, $\boldsymbol{T}^{\equiv}$, and $\boldsymbol{TS}^{\equiv}$, respectively. The $\boldsymbol{P}^{\equiv}$ and $\boldsymbol{T}^{\equiv}$ semantics form a dual pair, and the $\boldsymbol{R}^{\equiv}$ and $\boldsymbol{TS}^{\equiv}$ semantics are autodual.

Language of $L_{\mathrm{I}\perp}^{Q}$ is defined similarly to the language $L_{\perp}^{Q\equiv}$, but without the options related to $\equiv_{xy}$. The semantics $_\perp\boldsymbol{R}^{\mathrm{I}}$, $_\perp\boldsymbol{P}^{\mathrm{I}}$, $_\perp\boldsymbol{T}^{\mathrm{I}}$, $_\perp\boldsymbol{TS}^{\mathrm{I}}$ are specified in $L_{\mathrm{I}\perp}^{Q}$. $_\perp\boldsymbol{P}^{\mathrm{I}}$ and $_\perp\boldsymbol{T}^{\mathrm{I}}$ form a dual pair, while $_\perp\boldsymbol{R}^{\mathrm{I}}$ and $_\perp\boldsymbol{TS}^{\mathrm{I}}$ are autodual.

Language of L_{I}^{Q} is defined similarly to the language $L^{Q\equiv}$, but omitting the options related to $\equiv_{xy}$. The semantics $\boldsymbol{R}^{\mathrm{I}}$, $\boldsymbol{P}^{\mathrm{I}}$, $\boldsymbol{T}^{\mathrm{I}}$, and $\boldsymbol{TS}^{\mathrm{I}}$ are specified in L_{I}^{Q}. $\boldsymbol{P}^{\mathrm{I}}$ and $\boldsymbol{T}^{\mathrm{I}}$ form a dual pair, $\boldsymbol{R}^{\mathrm{I}}$ and $\boldsymbol{TS}^{\mathrm{I}}$ are autodual.

Language of $L_{\perp}^{Q}$ is defined similarly to the language $L_{\mathrm{I}\perp}^{Q}$, but without the options related to Ex. In $L_{\perp}^{Q}$ we specify the semantics $_\perp\boldsymbol{R}$, $_\perp\boldsymbol{P}$, $_\perp\boldsymbol{T}$, $_\perp\boldsymbol{TS}$. The $_\perp\boldsymbol{P}$ and $_\perp\boldsymbol{T}$ semantics form a dual pair, $_\perp\boldsymbol{R}$ and $_\perp\boldsymbol{TS}$ are autodual.

Language of L^{Q} is defined similarly to the language L_{I}^{Q}, but omitting the options related to Ex. In L^{Q} we specify the semantics $\boldsymbol{R}$, $\boldsymbol{P}$, $\boldsymbol{T}$, $\boldsymbol{TS}$. The $\boldsymbol{P}$ and $\boldsymbol{T}$ semantics form a dual pair, while $\boldsymbol{R}$ and $\boldsymbol{TS}$ are autodual.

In the languages $L_{\perp}^{Q}$ and L^{Q} every $\Phi\in Fr$ is a *CF*-formula, because the symbols of 0-ary compositions-constants are absent.

Language of $L_{\perp}^{Q=}$ is defined similarly to the language $L_{\perp}^{Q\equiv}$, except with the symbols $=_{xy}$ instead of $\equiv_{xy}$, and omitting the symbols Ex. The set of symbols of basic compositions is $\{\neg, \vee, R^{\bar v,\bar u}_{\bar x,\perp}, \exists x, =_{xy}\}$. The set Fr of formulas is determined as in the language $L_{\perp}^{Q\equiv}$, but omitting FI, and for the equality, we define

F=) $\{=_{xy} \mid x, y\in V\}\subseteq Fr$.

The function $\nu : Fr\to 2^V$ in the language $L_{\perp}^{Q=}$ is defined similarly to the language $L_{\perp}^{Q\equiv}$, but without $\nu(Ex)$, and for the equality we set $\nu(=_{xy}) = V\backslash\{x, y\}$. The language $L_{\perp}^{Q=}$ is interpreted in the same manner as $L_{\perp}^{Q\equiv}$. In $L_{\perp}^{Q=}$ we specify the semantics $_\perp\boldsymbol{R}^{=}$ and $_\perp\boldsymbol{P}^{=}$, as well as the semantics of monotone predicates $_\perp\boldsymbol{RM}^{=}$ and $_\perp\boldsymbol{PE}^{=}$.

Language of $L^{Q=}$ is defined similarly to the language $L_{\perp}^{Q=}$, except instead of the $R^{\bar v,\bar u}_{\bar x,\perp}$ we use $R^{\bar v}_{\bar x}$.

In $L^{Q=}$ we specify the semantics $\boldsymbol{R}^{=}$ and $\boldsymbol{P}^{=}$, and $\boldsymbol{RM}^{=}$ and $\boldsymbol{PE}^{=}$ as well.

Language of $L_{\downarrow\perp}^{Q}$ is defined similarly to the language $L_{\mathrm{I}\perp}^{Q}$, but instead of the symbols Ez we use the notation $\downarrow z$. The set Fr of formulas is determined as in the $L_{\mathrm{I}\perp}^{Q}$, except changing FI) to $\{\downarrow x \mid x\in V\}\subseteq Fr$. The function $\nu : Fr\to 2^V$ is defined similarly to the language $L_{\mathrm{I}\perp}^{Q}$, but changing $\nu(Ex)$ to $\nu(\downarrow x) = V\backslash\{x\}$. The language $L_{\downarrow\perp}^{Q}$ is interpreted in the same manner as $L_{\mathrm{I}\perp}^{Q}$. In $L_{\downarrow\perp}^{Q}$ we specify the semantics $_\perp\boldsymbol{R}^{\downarrow}$, $_\perp\boldsymbol{P}^{\downarrow}$, and $_\perp\boldsymbol{RM}^{\downarrow}$, $_\perp\boldsymbol{PE}^{\downarrow}$.

Language of $L_{\downarrow}{}^{Q}$ is defined similarly to the language $L_{\downarrow\perp}{}^{Q=}$, except instead of the symbols $R^{\bar{v},\bar{u}}_{\bar{x},\perp}$ we use $R^{\bar{v}}_{\bar{x}}$. The interpretation mapping is changed correspondingly. In $L_{\downarrow}{}^{Q}$ we specify the semantics $\boldsymbol{R}^{\downarrow}$ and $\boldsymbol{P}^{\downarrow}$, and $_{\perp}\boldsymbol{RM}^{\downarrow}$ and $_{\perp}\boldsymbol{PE}^{\downarrow}$ as well. The semantics of monotone predicates $_{\perp}\boldsymbol{RM}^{=}$, $_{\perp}\boldsymbol{PE}^{=}$, $\boldsymbol{RM}^{=}$, $\boldsymbol{PE}^{=}$, $_{\perp}\boldsymbol{RM}^{\downarrow}$, $_{\perp}\boldsymbol{PE}^{\downarrow}$, $\boldsymbol{RM}^{\downarrow}$, $\boldsymbol{PE}^{\downarrow}$ are not considered in this work.

The following definitions are the same for the languages described above. Predicate $\boldsymbol{J}(\Phi)$ is a value of the formula Φ in the interpretation $\boldsymbol{J}$; it will be denoted by Φ_J.

Formula Φ is *satisfiable in the interpretation* $\boldsymbol{J}$, or $\boldsymbol{J}$-satisfiable, if the predicate Φ_J is satisfiable.

Formula Φ is *satisfiable in the semantics* α, if Φ is satisfiable in some $\boldsymbol{J}\in\alpha$.

Formula Φ is *irrefutable* (*partially true*) *in the interpretation* $\boldsymbol{J}$, or $\boldsymbol{J}$-irrefutable (denoted by $\boldsymbol{J}\,|\!=\Phi$), if the predicate Φ_J is irrefutable.

Formula Φ is *irrefutable in the semantics* α (denoted by ${}^{\alpha}|\!=\Phi$),), if $\boldsymbol{J}\,|\!=\Phi$ for every $\boldsymbol{J}\in\alpha$.

Formula Φ is *totally true in the interpretation* $\boldsymbol{J}$ (denoted by $\boldsymbol{J}\,|\!=_{id}\Phi$), if $\Phi_J = T$.

Formula Φ is *totally true in the semantics* α (denoted by ${}^{\alpha}|\!=_{id}\Phi$), if $\boldsymbol{J}\,|\!=_{id}\Phi$ for every $\boldsymbol{J}\in\alpha$.

If α is referred by default, we can write $|\!=$ and $|\!=_{id}$ instead of ${}^{\alpha}|\!=$ and ${}^{\alpha}|\!=_{id}$.

Statement 14. $\boldsymbol{J}\,|\!=_{id}\Phi \Rightarrow \boldsymbol{J}\,|\!=\Phi$; ${}^{\alpha}|\!=_{id}\Phi \Rightarrow {}^{\alpha}|\!=\Phi$.

Example 1. $|\!=_{id} Ex\vee\neg Ex$, $|\!=_{id}\exists xEx$, $|\!=_{id}\neg\exists x\neg Ex$, $|\!=_{id}\equiv_{xx}$.

Let us call *constant* the formulas that are always interpreted as the constant predicates: $_{T}$-*formulas* (interpreted as T), $_{F}$-*formulas* (interpreted as F), and $_{\perp}$-*formulas* (interpreted as $\perp$).

Example 2. $Ex\vee\neg Ex$, $\exists xEx$, $\exists x\downarrow x$, $\exists x =_{xx}$ and $\equiv_{xx}$ are $_{T}$-formulas; $R^{\bar{v},\bar{u},z}_{\bar{x},\perp,\perp}(Ez)$, $\exists x\neg Ex$, $\exists x\neg\downarrow x$, $\exists z\neg=_{zz}$ and $\neg\equiv_{xx}$ are $_{F}$-formulas; $R^{\bar{v},\bar{u},z}_{\bar{x},\perp,\perp}(=_{zy})$, $R^{z}_{\perp}(=_{zy})$, $R^{\bar{v},\bar{u},z}_{\bar{x},\perp,\perp}(=_{zz})$, $R^{z}_{\perp}(=_{zz})$, $R^{\bar{v},\bar{u},z}_{\bar{x},\perp,\perp}(\downarrow z)$, $R^{z}_{\perp}(\downarrow z)$ are $_{\perp}$-formulas.

If ϑ is a $_{T}$-formula, then ${}^{\alpha}|\!=_{id}\vartheta$ (α is of type R or P).

Let us also define *partially constant formulas*. Such are $_{pT}$-*formulas* (for them, $F(\Phi_J)=\varnothing$ for all $\boldsymbol{J}$) and $_{pF}$-*formulas* (for them, $T(\Phi_J)=\varnothing$ for all $\boldsymbol{J}$). For $_{pT}$-formulas Φ, we have ${}^{\alpha}|\!=\Phi$ (here α is of type R or P).

Example 3. The symbols $\downarrow x$ and $=_{zz}$ are $_{pT}$-formulas, while $\neg\downarrow x$ and $\neg=_{zz}$ are $_{pF}$-formulas.

4. Logical consequence relations in PCNL languages

Depending on relations between the truth and falsity domains of predicates, a series of logical consequence relations on the set of formulas are introduced in work (Nikitchenko, Shkilniak, & Shkilniak, 2016). The "natural" consequence relations for a pair of formulas Φ and Ψ under a fixed interpretation $\boldsymbol{J}$ are defined as follows:

- truth, or T-consequence ${}_{J}|\!=_{T}$: $\Phi\ {}_{J}|\!=_{T}\Psi \Leftrightarrow T(\Phi_J)\subseteq T(\Psi_J)$;
- falsity, or F-consequence ${}_{J}|\!=_{F}$: $\Phi\ {}_{J}|\!=_{F}\Psi \Leftrightarrow F(\Psi_J)\subseteq F(\Phi_J)$;
- strong, or TF-consequence ${}_{J}|\!=_{TF}$: $\Phi\ {}_{J}|\!=_{TF}\Psi \Leftrightarrow \Phi\ {}_{J}|\!=_{T}\Psi$ and $\Phi\ {}_{J}|\!=_{F}\Psi$;
- irrefutability, or IR-consequence ${}_{J}|\!=_{IR}$: $\Phi\ {}_{J}|\!=_{IR}\Psi \Leftrightarrow T(\Phi_J)\cap F(\Psi_J)=\varnothing$;
- dual to IR, or DI-consequence ${}_{J}|\!=_{DI}$: $\Phi\ {}_{J}|\!=_{DI}\Psi \Leftrightarrow F(\Phi_J)\cup T(\Psi_J)={}^{V}\!A$.

These relations in the interpretation $\boldsymbol{J}$ induce the corresponding equivalence relations ${}_{J}{\sim}_{T}$, ${}_{J}{\sim}_{F}$, ${}_{J}{\sim}_{TF}$, ${}_{J}{\sim}_{IR}$, ${}_{J}{\sim}_{DI}$ in the interpretation $\boldsymbol{J}$:

$$\Phi\ {}_{J}{\sim}_{*}\Psi,\ \text{if}\ \Phi\ {}_{J}|\!=_{*}\Psi\ \text{and}\ \Psi\ {}_{J}|\!=_{*}\Phi.$$

The relation ${}_{J}{\sim}_{TF}$ is special: $\Phi\ {}_{J}{\sim}_{TF}\Psi$ means $\Phi_J=\Psi_J$, i. e. it is the same predicate.

The *logical* τ-*consequence* relation in the semantics α is specified in the form:

$$\Phi\ {}^{\alpha}|\!=\tau\,\Psi,\ \text{if}\ \Phi\ {}_{J}|\!=\tau\,\Psi\ \text{for all}\ \boldsymbol{J}\in\alpha.$$

The *logical* τ-*equivalence* relation in the semantics α is specified in the form:

$$\Phi\ {}^{\alpha}{\sim}_{*}\Psi,\ \text{if}\ \Phi\ {}^{\alpha}|\!=_{*}\Psi\ \text{and}\ \Psi\ {}^{\alpha}|\!=_{*}\Phi.$$

For each of the logics L^{Q}, $L_{\perp}{}^{Q}$, $L_{\downarrow}{}^{Q}$, $L_{\downarrow\perp}{}^{Q}$, $L^{Q\equiv}$ and $L_{\perp}{}^{Q\equiv}$, we distinguished 4 types of semantics: R, P, T, TS, resulting in 20 consequence relations in each of these logics. However, as noted in (Nikitchenko, Shkilniak, O., & Shkilniak, S., 2016), some of these relations coincide, while others are degenerate. Taking into account the duality of P-semantics and T-semantics and the degeneracy of TS-semantics, we consider only semantics of types R and P. This results in 5 distinct non-degenerate logical consequence relations: ${}^{P}{}_{*}|\!=_{IR}$, ${}^{P}{}_{*}|\!=_{T}$, ${}^{P}{}_{*}|\!=_{F}$, ${}^{P}{}_{*}|\!=_{TF}$; ${}^{R}{}_{*}|\!=_{TF}$.

Graphically, the relationships between them can be represented as follows (using $\rightarrow$ instead $\subset$):

$$\begin{array}{ccccccc} & & & & {}^{P}{}_{*}|\!=_{T} & & \\ & & & \nearrow & & \searrow & \\ {}^{R}{}_{*}|\!=_{TF} & \rightarrow & {}^{P}{}_{*}|\!=_{TF} & & & & {}^{P}{}_{*}|\!=_{IR} \\ & & & \searrow & & \nearrow & \\ & & & & {}^{P}{}_{*}|\!=_{F} & & \end{array}$$

In R-semantics of the logics L^{Q}, $L_{\perp}{}^{Q}$, $L_{\downarrow}{}^{Q}$, $L_{\downarrow\perp}{}^{Q}$, $L^{Q\equiv}$ and $L_{\perp}{}^{Q\equiv}$, the relations of types T, F, and TF coincide.

Statement 15. 1*) There do not exist CF-formulas* Φ *and* Ψ *such that* $\Phi\ {}^{R}{}_{*}|\!=_{IR}\Psi$;

2) there do not exist CF-formulas Φ *and* Ψ *such that* $\Phi\ {}^{R}{}_{*}|\!=_{DI}\Psi$ *or* $\Phi\ {}^{P}{}_{*}|\!=_{DI}\Psi$.

Corollary 1. *The relations* ${}^{R}_{*}|\!=_{DI}$, ${}^{P}_{*}|\!=_{DI}$, *and* ${}^{R}_{*}|\!=_{IR}$ *are degenerate.*

In the logics $L_{\perp}^{Q=}$, $L^{Q=}$, $L_{\downarrow\perp}^{Q}$, and $L_{\downarrow}^{Q}$ we specified 2 semantics of types R and P (not including semantics of monotone predicates, not considered here). This results in 10 logical consequence relations for each of the logics. Some of these relations coincide, while others are degenerate. Specifically, the relations ${}^{R}_{*}|\!=_{DI}$, ${}^{P}_{*}|\!=_{DI}$, ${}^{R}_{*}|\!=_{IR}$ are among them (see Statement 15).

In $L_{\perp}^{Q=}$, $L^{Q=}$, $L_{\downarrow\perp}^{Q}$ and $L_{\downarrow}^{Q}$, the relations of types T, F, TF in P- and R-semantics are significantly different:

Statement 16. *1)* $\Phi\ {}^{\alpha}|\!=_{F} \downarrow x$ *and* $\Phi\ {}^{\alpha}|\!=_{IR} = \downarrow x$ *for arbitrary* $\Phi \in Fr$ *and* $x \in V$*;*

2) $\neg\downarrow x\ {}^{\alpha}|\!=_{T} \Phi$ *and* $\neg\downarrow x\ {}^{\alpha}|\!=_{IR} \Phi$ *for any* $\Phi \in Fr$ *and* $x \in V$*;*

3) $\exists x \downarrow x\ {}^{\alpha}|\!\neq_{T} \downarrow y$ *and* $\neg\downarrow y\ {}^{\alpha}|\!\neq_{F} \neg\exists x \downarrow x$*;*

4) $\downarrow x\ {}^{\alpha}\!\sim_{F} \downarrow y$ *and* $\downarrow x\ {}^{\alpha}\!\sim_{IR} \downarrow y$ *for any* $x, y \in V$.

Therefore, for each of $L_{\perp}^{Q=}$, $L^{Q=}$, $L_{\downarrow\perp}^{Q}$, $L_{\downarrow}^{Q}$ we obtain 7 distinct non-degenerate logical consequence relations:

$${}^{P}_{*}|\!=_{IR},\ {}^{P}_{*}|\!=_{T},\ {}^{P}_{*}|\!=_{F},\ {}^{P}_{*}|\!=_{TF},\ {}^{R}_{*}|\!=_{T},\ {}^{R}_{*}|\!=_{F},\ {}^{R}_{*}|\!=_{TF}.$$

The relationships between them can be represented as follows

$$\begin{array}{ccccc}
 & {}^{R}_{*}|\!=_{T} & \rightarrow & {}^{P}_{*}|\!=_{T} & \\
\nearrow & & \nearrow & & \searrow \\
{}^{R}_{*}|\!=_{TF} & \rightarrow & {}^{P}_{*}|\!=_{TF} & & {}^{P}_{*}|\!=_{IR} \\
\searrow & & \searrow & & \nearrow \\
 & {}^{R}_{*}|\!=_{F} & \rightarrow & {}^{P}_{*}|\!=_{F} &
\end{array}$$

The considered logical consequence relations for a pair of formulas are reflexive and transitive, while the logical equivalence relations are reflexive, transitive, and symmetric.

For relations of types IR and TF, the equivalence theorem holds, but it fails for relations of types T and F (Nikitchenko, Shkilniak, O., & Shkilniak, S., 2016).

Theorem 4 *(equivalence). Let a* Φ' *be obtained from formula* Φ *by replacing some occurrences* $\Phi_1, ..., \Phi_n$ *with* $\Psi_1, ..., \Psi_n$*. If* $\Phi_1\ {}^{\alpha}\!\sim_{*} \Psi_1, ..., \Phi_n\ {}^{\alpha}\!\sim_{*} \Psi_n$*, then* $\Phi\ {}^{\alpha}\!\sim_{*} \Phi'$*.*

The aforementioned properties of compositions induce similar properties for formulas in their respective languages, denoted as ${}^{*}\!\sim_{TF}$ (Nikitchenko, Shkilniak, O., & Shkilniak, S., 2016).

Statement 17. *The transitivity of weak equality is violated for relations* ${}^{P=}|\!=_{F}$ *and* ${}^{P=}|\!=_{T}$*:*

$$=_{xy} \& =_{yz}\ {}^{P=}|\!\neq_{F}\ =_{xy} \& =_{yz} \& =_{xz} \text{ and } \neg =_{xy} \vee \neg =_{yz} \vee \neg =_{xz}\ {}^{P=}|\!\neq_{T}\ \neg =_{xy} \vee \neg =_{yz}.$$

For $d = [x \mapsto a, z \mapsto b]$ we have $d \notin F(=_{xy}) \cup F(=_{yz}) = F(=_{xy} \& =_{yz})$; simultaneously, $d \in F(=_{xz}) \subseteq F(=_{xy} \& =_{yz} \& =_{xz})$. Next, we take into consideration that $\Phi\ {}^{P=}|\!=_{F} \Psi \Leftrightarrow \neg\Psi\ {}^{P=}|\!=_{T} \neg\Phi$.

Given that the transitivity of weak equality is violated for ${}^{P=}|\!=_{F}$ and ${}^{P=}|\!=_{T}$, thus it will be violated for stronger relations of types F, T, TF in $L^{Q=}$ and $L_{\perp}^{Q=}$.

Therefore, in the logics $L^{Q=}$ and $L_{\perp}^{Q=}$, all the T- F- and TF-relations are *incorrect*.

Corollary 2. *In the logics* $L^{Q=}$ *and* $L_{\perp}^{Q=}$*, the only non-degenerate and correct relations are* ${}^{P=}|\!=_{IR}$ *and* ${}^{P=}_{\perp}|\!=_{IR}$*.*

Discussion and conclusions

In this paper we studied the semantic properties of new classes of program-oriented logical formalisms – pure first-order logics of quasiary predicates, or PCNL. Depending on the use of traditional or extended renomination compositions, the presence and type of predicates-indicators, and equality predicates, corresponding classes of such logics were distinguished. New classes of PCNL with partial predicates-indicators were proposed. The languages of the introduced classes of PCNL were described, and a number of logical consequence relations for these languages were defined. The characteristics of these relations were investigated, and the relationships between them were specified.

A detailed description of PCNL with partial predicates-indicators and the study of logical consequence relations for sets of formulas are planned for future works.

References

Abramsky, S., Gabbay, D. M., & Maibaum, T. S. E. (Eds). (1993–2000). *Handbook of Logic in Computer Science* (Vol. 1–5). Oxford University Press.

Barwise, J. (Ed). (1977). *Handbook of Mathematical Logic* (Part 1–4). North-Holland Publishing Company.

Bjorner, D., & Henson, M. C. (Eds). (2008). *Logics of Specification Languages, EATCS Series.* Springer.

Huth, M., and Ryan, M. (2004). *Logic in Computer Science. Second Edition.* Cambridge University Press.

Kleene, S. C. (2013). *Mathematical Logic*. Dower Publications.

Nikitchenko, M., Shkilniak, O., & Shkilniak, S. (2016). Pure first-order logics of quasiary predicates. *Problems in Programming,* 2–3, 73–86 [in Ukrainian]. [Нікітченко, М., Шкільняк, О., Шкільняк, С. (2016). Чисті першопорядкові логіки квазіарних предикатів. *Проблеми програмування*, 2–3, 73–86].

Shoenfield, J. (1967). *Mathematical Logic.* Addison-Wesley Publishing company.

Shkilniak, S. (2019). First-order composition-nominative logics with predicates of weak equality and of strong equality. *Problems in Programming,* 3, 28–44. [in Ukrainian]. [Шкільняк, С. (2019). Першопорядкові композиційно-номінативні логіки з предикатами слабкої та строгої рівності. *Проблеми програмування*, 3, 28–44].

Shkilniak, O., & Shkilniak, S. (2023). First-Order Sequent Calculi of Logics of Quasiary Predicates with Extended Renominations and Equality. *Proceedings of UkrPROG'2022, Kyiv, Ukraine. CEUR Workshop Proceedings (CEUR-WS.org)*, 3–18.

Отримано редакцією журналу / Received: 05.07.24
Прорецензовано / Revised: 15.11.24
Схвалено до друку / Accepted: 26.11.24

Степан ШКІЛЬНЯК, д-р фіз.-мат. наук, проф.
ORCID ID: 0000-0003-8624-5778
e-mail: ss.sh@knu.ua
Київський національний університет імені Тараса Шевченка, Київ, Україна

РІЗНОВИДИ ЧИСТИХ ПЕРШОПОРЯДКОВИХ ЛОГІК ЧАСТКОВИХ КВАЗІАРНИХ ПРЕДИКАТІВ

***В основі різноманітних логічних систем, які з великим успіхом використовують в інформатиці та програмуванні, зазвичай лежить класична логіка предикатів. Водночас класична логіка має принципові обмеження, що ускладнюють її застосування. З огляду на це вельми важливою стає задача розроблення нових, програмно-орієнтованих логік; це засвідчує актуальність пропонованого дослідження. Важливим класом таких логік є композиційно-номінативні логіки часткових квазіарних предикатів. Представлена робота продовжує дослідження чистих першопорядкових композиційно-номінативних логік (ЧКНЛ). Нові класи ЧКНЛ отримано шляхом уведення часткових предикатів-індикаторів. Спеціальні предикати-індикатори визначають наявність у вхідних даних компоненти з відповідним предметним іменем, що необхідно враховувати під час елімінації кванторів у логіках немонотонних предикатів. Виділено два типи цих предикатів: тотальні Ez, що встановлюють наявність чи відсутність компоненти зі заданим іменем z, та часткові, що засвідчують лише наявність такої компоненти. Запропоновано часткові предикати-індикатори ↓z та відповідні класи ЧКНЛ. Залежно від наявності та типу предикатів-індикаторів, предикатів рівності (слабкої $=_{xy}$ чи строгої $\equiv_{xy}$), а також від використання традиційної чи розширеної реномінації, визначено низку класів ЧКНЛ. На базовому рівні маємо логіки L^Q із традиційними реномінаціями та логіки $L_{\perp}^Q$ із розширеними реномінаціями. Розширення цих логік за
допомогою предикатів-індикаторів ↓z надає нові логіки $L_{\downarrow}^Q$ та $L_{\downarrow\perp}^Q$, а їхнє розширення за допомогою предикатів-індикаторів Ez – нові логіки L_{I}^Q та $L_{I\perp}^Q$. Подальше розширення таких логік за допомогою предикатів рівності $=_{xy}$ та $\equiv_{xy}$ надає відповідно логіки $L^{Q=}$, $L_{\perp}^{Q=}$ та $L^{Q\equiv}$, $L_{\perp}^{Q\equiv}$. Описано базові композиції цих логік, наведено їхні основні властивості. Описано мови введених класів L^Q, визначено низку відношень логічного наслідку в цих мовах. Досліджено властивості цих відношень та наведено взаємовідношення між ними.***

Ключові слова: ***логіка, частковий предикат, рівність, логічний наслідок.***

Автор заявляє про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The author declares no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.