

КОМПОЗИЦІЙНО-НОМІНАТИВНІ ЛОГІКИ НАД ІЄРАРХІЧНИМИ ДАНИМИ

М.С. Нікітченко, С.С. Шкільняк

Київський національний університет імені Тараса Шевченка,
01601, Київ, вул. Володимирська, 60
Тел.: (044) 259 0519
E-mail: ttp@unicyb.kiev.ua

Будуються нові логіки, які дозволяють більш адекватно описувати властивості предикатів та функцій, визначених над ієрархічними даними. Характерною властивістю цих логік є використання в їх мовах складних імен. Вивчаються семантичні властивості таких логік та визначаються відповідні числення секвенційного типу.

New logics more adequate for property descriptions of functions and predicates over hierarchical data are constructed. A characteristic feature of such logics is use of composite names in their languages. Semantic properties of such logics are investigated, corresponding sequent calculi are defined.

Вступ

Математична логіка є одним з наріжних каменів, на яких ґрунтується теорія програмування. Це означає, що поняття і методи логіки знаходять своє найширше використання у дослідженнях програм, що яскраво підтверджують багатомні видання із застосувань логіки в інформатиці (див., напр., [1]). Таке використання логіки має і зворотній вплив, а саме, теорія програмування вимагає змін в логіці щоб зробити її більш близькою та адекватною до програмування. Одним з підходів у цьому напрямку є композиційно-номінативний підхід (КНП) до побудови моделей програм та орієнтованих на них логік [2–5]. Підхід задає принципи визначення та дослідження формальних мов програм та логік. Суть підходу, стисло кажучи, полягає в наступному:

- семантичний аспект мови є провідним, синтаксичний – похідним; тому спочатку визначаються та досліджуються семантичні аспекти, а потім – синтаксичні;
- семантичні аспекти уточнюються за допомогою композиційно-номінативних систем, які визначають структури даних, функцій (зокрема, предикатів) та композицій; такі системи можна подавати у вигляді двох семантичних алгебр – алгебри даних та алгебри функцій (предикатів);
- дані уточнюються як номінативні – ієрархічні дані, побудовані на базі відношення ім'я \rightarrow значення;
- основні структури даних мов програмування можна подати як конкретизації номінативних даних [6];
- властивості семантичних алгебр індукують синтаксичні аспекти мов;
- побудова мов іде в напрямку від абстрактного до конкретного, що приводить до мов різного рівня абстракції та загальності.

Для розбудови нових типів логік на підставі КНП слід проаналізувати структуру класичної логіки. Такий аналіз проведено в [4]. Його результати (стосовно теми статті) можна подати таким чином:

- у мові логіки виділяються два типи символів: логічні, до яких належать символи логічних зв'язок і кванторів, та дескриптивні (предметні), до яких належить символи предметних змінних, предикатів та функцій;
- предметна область подається алгебраїчною системою, носієм якої є множина предметних констант (атомів), а операціями – предметні функції та предикати; таку систему можна розглядати як алгебру даних; інтерпретація дескриптивних символів задається окремо для кожної предметної області;
- логічні символи мають фіксовану інтерпретацію, незалежну від вибору предметної області, тому в КНП логічні символи інтерпретуються як композиції, які є операціями алгебри предикатів.

Таким чином, логіки першого порядку фактично засновані на алгебрах атомарних, неструктурованих даних. Тому при застосуванні класичної логіки першого порядку до програм над складними, ієрархічними даними виникає певна невідповідність між мовою логіки, яка дозволяє працювати лише з атомарними даними, та мовою програм, яка працює з ієрархічними даними. Для подолання такої невідповідності класична логіка має певні механізми, які дозволяють промоделювати складні дані через атомарні. Але при цьому значні зусилля витрачаються на саме моделювання, а також на дослідження його властивостей, які потрібно подати за допомогою певної аксіоматики, визначити її коректність, повноту тощо. Цих складнощів можна уникнути, побудувавши логіку, безпосередньо засновану на алгебрах ієрархічних даних. Це і є метою даної статті.

Стаття побудована наступним чином: у першому розділі вводяться поняття на визначення ієрархічних даних, у другому – операції над такими даними, третій розділ присвячено композиціям. У четвертому розділі описуються семантичні моделі та відповідні мови логік, п'ятий розділ присвячено визначенню секвенційного числення для однієї з побудованих логік. Поняття, які тут не визначаються, будемо тлумачити в сенсі роботи [5].

1. Основні поняття та визначення

Основним поняттям логіки з семантичного погляду є поняття предиката.

У загальному випадку під предикатом на множині D будемо розуміти довільну часткову функцію вигляду $P : D \rightarrow \{T, F\}$. У даній роботі будемо розглядати однозначні часткові предикати.

Областю істинності та областю хибності однозначного предиката P на D назвемо множини

$$T(P) = \{d \in D \mid P(d) = T\} \quad \text{та} \quad F(P) = \{d \in D \mid P(d) = F\}.$$

Предикат P на D неспростовний, або частково істинний, якщо $F(P) = \emptyset$.

Номінативні дані можна трактувати як ієрархічно побудовані об'єкти, в основі яких лежить бінарне відношення ім'я \mapsto значення. Найпростішим випадком номінативних даних є однорівневі номінативні дані, які звичайно називають [5] іменними множинами.

Нехай A та V – непорожні множини, які трактуємо як множину базових імен та множину базових значень.

V -іменною множиною (V -IM) над A називають [5] довільну однозначну функцію $\delta : V \rightarrow A$.

V -IM подаємо у вигляді $[v_1 \mapsto a_1, \dots, v_n \mapsto a_n, \dots]$. Тут $v_i \in V$, $a_i \in A$, $v_i \neq v_j$ при $i \neq j$.

У загальному випадку номінативних даних іменування можна продовжити на складні дані, а не тільки базові. На відміну від [6], при побудові складніших номінативних даних із простіших тут не будемо використовувати кроки побудови за типом множини.

Множину ієрархічних номінативних даних $ND(V, A)$ над множиною базових імен V та множиною базових значень A визначаємо індуктивно наступним чином.

$$ND_0(V, A) = A \cup \{\emptyset\} \text{ – номінативні дані рангу } 0;$$

$$ND_{k+1}(V, A) = A \cup (V \xrightarrow{n} ND_k(V, A)) \text{ – номінативні дані рангу } \leq k+1.$$

$$\text{Тоді } ND(V, A) = \bigcup_{k \geq 0} (V \xrightarrow{n} ND_k(V, A)).$$

Тут $V \xrightarrow{n} ND_k(V, A)$ – множина всіх однозначних відображень із V в $ND_k(V, A)$.

Однозначність таких відображень гарантує однозначність іменування.

Визначимо також множину $HD(V, A) = ND(V, A) \setminus A$ іменованих ієрархічних номінативних даних.

Для складного імені $u = y_1.y_2.\dots.y_n$ запис $d(u)$ буде означати $(\dots(d(y_1)(y_2))\dots(y_n))$. Надалі для більшої виразності замість $d(u)$ будемо в стилі функцій розіменування писати $u(d)$. Таке $u(d)$ – це ієрархічне дане, що є компонентою даного d , яка поіменована в d іменем u . Для складного імені $u = y_1.y_2.\dots.y_n$ запис $u(d)$ означатиме $y_n(\dots(y_2(y_1(d))))$.

Компоненту ієрархічного даного, подану у вигляді $x \mapsto u(d)$, у випадку невизначеності значення імені u (позначаємо $u(d) \uparrow$) будемо опускати.

Ієрархічні номінативні дані можна подавати різними способами. Будемо використовувати тут текстовий, або лінійний спосіб, та графічний спосіб.

Приклад 1. Розглянемо ієрархічне дане, текстовий запис якого має наступний вигляд:

$$[x \mapsto [x \mapsto [x \mapsto 0, y \mapsto 2, z \mapsto 0], z \mapsto 1], y \mapsto 0, z \mapsto [x \mapsto 1, y \mapsto [x \mapsto 1, y \mapsto 2], z \mapsto 3]].$$

Таке дане можна записати в структурованішому вигляді:

$$\begin{aligned} & [x \mapsto [x \mapsto [x \mapsto 0, \\ & \quad y \mapsto 2, \\ & \quad z \mapsto 0], \\ & \quad z \mapsto 1], \\ & y \mapsto 0, \\ & z \mapsto [x \mapsto 1, \\ & \quad y \mapsto [x \mapsto 1, \\ & \quad \quad y \mapsto 2], \\ & \quad z \mapsto 3]]. \end{aligned}$$

При графічному способі ієрархічне дане подаємо у вигляді орієнтованого дерева, ребра якого помічаємо іменами, а кінцеві вершини (листки) помічаємо базовими значеннями. Однозначність іменування означає, що всі ребра, які виходять з однієї вершини, помічаються різними іменами.

Фактично таке дерево прочитується із другого запису: корінь дерева – зліва, листки – справа.

Ієрархічні дані можна трактувати як номінативні дані зі складними іменами – елементами множини V^+ . Інакше кажучи, складними іменами є непорожні слова в алфавіті V , вони утворюються конкатенацією базових імен при русі по дереву з кореня до листків. Операцію конкатенації будемо позначати символом ”.”.

Приклад 2. Ієрархічне дане прикладу 1 можна перетворити у такі дані зі складними іменами:

$$\begin{aligned} & [x.x \mapsto [x \mapsto 0, y \mapsto 2, z \mapsto 0], x.z \mapsto 1, y \mapsto 0, z.x \mapsto 1, z.y \mapsto [x \mapsto 1, y \mapsto 2], z.z \mapsto 3]; \\ & [x.x.x \mapsto 0, x.x.y \mapsto 2, x.x.z \mapsto 0, x.z \mapsto 1, y \mapsto 0, z.x \mapsto 1, z.y.x \mapsto 1, z.y.y \mapsto 2, z.z \mapsto 3]. \end{aligned}$$

Останнє, 1-рівневе подання – це, фактично, іменна множина зі складними іменами.

Форму подання ієрархічних даних у вигляді іменної множини зі складними іменами назвемо лінійною нормальною формою (ЛНФ).

Зважаючи на однозначність іменування для кожного ієрархічного даного всі (складні) імена його лінійної нормальної форми мусять бути різними, більше того, вони будуть непорівнянними (визначення див. далі).

Формально кажучи, ієрархічне дане прикладу 1 та ієрархічні дані прикладу 2 мають різну структуру іменування, проте в даній роботі будемо абстрагуватись від такої структури і вважатимемо всі наведені в цих прикладах ієрархічні дані еквівалентними, а дослідження таких даних та предикатів над ними будемо проводити з точністю до еквівалентності ієрархічних даних.

Ієрархічні дані d_1 та d_2 назвемо еквівалентними, якщо d_1 та d_2 мають однакові ЛНФ.

Приймаючи таку еквівалентність, замість $x.u \mapsto \alpha$ можна писати $x \mapsto u \mapsto \alpha$ і навпаки.

Для складних імен природним чином вводиться поняття префікса. Префіксом слова $u \in V^+$ назвемо кожне слово x таке, що $u = x.u$ для деякого $u \in V^*$. Якщо при цьому $u \neq x$, то x назвемо власним префіксом слова u .

Будемо писати $x \leq u$, якщо x є префіксом слова u . Якщо x є власним префіксом слова u , то пишемо $x < u$.

Слова x та u назвемо префікс-порівнянними, або просто порівнянними, якщо $x \leq u$ або $u \leq x$; інакше такі слова назвемо префікс-непорівнянними, або просто непорівнянними.

Будемо писати $x <> u$, якщо x та u – порівнянні; пишемо $x \div u$, якщо x та u – непорівнянні.

Поширимо непорівнянність на множини імен: писатимемо $X \div Y$, якщо $x \div y$ для всіх $x \in X$ та $y \in Y$.

Для складних імен як слів можна ввести операцію $_x/$ ділення на ім'я x зліва:

$_x/u$ – це ім'я y таке, що $u = x.y$, якщо x є префіксом u , інакше $_x/u$ невизначене.

Для ієрархічних даних визначимо множини головних та термінальних імен.

Неформально кажучи, головне ім'я – це складне ім'я ієрархічного даного, які прочитуємо при русі по дереву від кореня. Якщо при цьому добираємось до листка, то таке ім'я – термінальне.

Множиною головних імен ієрархічного даного d назвемо множину $hn(d) = \{u \mid u(d) \downarrow\}$.

Множиною термінальних імен ієрархічного даного d назвемо множину $tn(d) = \{u \mid u(d) \downarrow \in A\}$.

Інакше кажучи, $tn(d)$ складається з усіх імен лінійної нормальної форми даного d , потрактованої як ІМ; $hn(d)$ складається з усіх імен лінійної нормальної форми даного d та їх префіксів.

Ієрархічні дані d_1 та d_2 назвемо диз'юнктними, якщо для довільних $x \in tn(d_1)$ та $y \in tn(d_2)$ маємо $x \div y$.

Надалі “+” будемо позначати операцію об'єднання двох диз'юнктних ієрархічних даних.

Тоді можна записати наступну властивість ієрархічних даних: $[x \mapsto (\alpha + \beta), \varphi] = [x \mapsto \alpha, x \mapsto \beta, \varphi]$.

2. Операції над ієрархічними даними

Над ієрархічними номінативними даними введемо операції видалення компонент із заданою умовою на імена, накладення та реномінації.

Параметричну операцію $|-_x$ видалення компонент з префікс-іменем x задамо так:

$|-_x(d) = [u \mapsto \alpha \in d \mid x \text{ не є префіксом } u]$.

Узагальнимо операцію $|-_x$ до операції $|-_{x_1, \dots, x_n}$:

$|-_{x_1, \dots, x_n}(d) = [u \mapsto \alpha \in d \mid \text{жодне з імен } x_1, \dots, x_n \text{ не є префіксом } u]$.

Для ієрархічних даних в ЛНФ це означає видалення компонент, для імен яких принаймі одне з x_1, \dots, x_n є префіксом.

Окрім операції $|-_{x_1, \dots, x_n}$ видалення компонент за умовою на префікси імен, задамо операції видалення компонент за більш сильною умовою префікс-порівнянності імен. Необхідність їх введення зумовлена тим, що не завжди додавання до $|-_x(d)$ компоненти з іменем x є можливим через порушення однозначності іменування.

Приклад 3. Нехай $d = [x \mapsto 1, z.x \mapsto 0, z.y \mapsto 1]$. Тоді $|-_{x,y}(d) = d$, але додавання до d компоненти з іменем $x.y$ не є коректним.

Параметричну операцію $||_-_x$ видалення компонент з іменами, порівнянними з іменем x , задамо так:

$||_-_x(d) = [u \mapsto a \in d \mid x \text{ та термінальне } u \text{ не є порівнянними}]$.

Узагальнимо операцію $||_-_x$ до операції $||_-_{x_1, \dots, x_n}$:

$||_-_{x_1, \dots, x_n}(d) = [u \mapsto a \in d \mid \text{жодне з імен } x_1, \dots, x_n \text{ не є порівнянним з термінальним іменем } u]$.

Для базових даних $a \in A$ будемо вважати $||_-_{x_1, \dots, x_n}(a)$ невизначеним.

Замість $||_-_x(d)$ та $||_-_{x_1, \dots, x_n}(d)$ будемо надалі писати $d ||_-_x$ та $d ||_-_{x_1, \dots, x_n}$.

Приклад 4. Нехай d – ієрархічне дане прикладу 1. Тоді:

$||_-_x(d) = [y \mapsto 0, z \mapsto [x \mapsto 1, y \mapsto [x \mapsto 1, y \mapsto 2], z \mapsto 3]]$; $||_-_{x,x,y,z,z,z}(d) = [x \mapsto [z \mapsto 1], z \mapsto [x \mapsto 1, y \mapsto [x \mapsto 1, y \mapsto 2]]]$.

При поданні d в ЛНФ маємо:

$||_-_x(d) = [y \mapsto 0, z.x \mapsto 1, z.y.x \mapsto 1, z.y.y \mapsto 2, z.z \mapsto 3]$; $||_-_{x,x,y,z,z,z}(d) = [x.z \mapsto 1, z.x \mapsto 1, z.y.x \mapsto 1, z.y.y \mapsto 2]$.

Неважко переконатись, що додавання до $||_-_{x_1, \dots, x_n}(d)$ компонент з іменами x_1, \dots, x_n є коректним, воно не порушує однозначність іменування.

Надалі для спрощення запису при використанні символу операції “+” будемо опускати зрозумілі з контексту дужки “{” та “}”. Наприклад, замість $d \parallel_{-u} + [u \mapsto u(d) \parallel_{-v}]$ будемо писати $d \parallel_{-u} + u \mapsto u(d) \parallel_{-v}$.

Розглянемо властивості операції $\parallel_{-x_1, \dots, x_n}$. Для такої операції, зокрема, маємо “поглинання” конкретнішого імені до імені загальнішим, яке є префіксом конкретнішого імені.

Твердження 1. Для ієрархічних даних маємо:

- 1) нехай $z \leq u$; тоді $d \parallel_{-z, u, x_1, \dots, x_n} = d \parallel_{-z, x_1, \dots, x_n}$;
- 2) $(d \parallel_{-x}) \parallel_{-x, y} = (d \parallel_{-x, y}) \parallel_{-x} = d \parallel_{-x}$;
- 3) $(d_1 + d_2) \parallel_{-u} = d_1 \parallel_{-u} + d_2 \parallel_{-u}$;
- 4) $d \parallel_{-u, v} = d \parallel_{-u} + [u \mapsto u(d) \parallel_{-v}]$;
- 5) $d \supseteq d \parallel_{-u} + u \mapsto u(d)$;
- 6) для простого імені $x \in V$ маємо $d = d \parallel_{-x} + x \mapsto x(d)$.

Приклад 5. Для складного імені u властивість $d = d \parallel_{-u} + u \mapsto u(d)$ може не виконуватись. Справді, нехай $d = [u \mapsto 1, u, x \mapsto 1, u, y \mapsto 2]$. Тоді $d \parallel_{-u, v} = [u, x \mapsto 1, u, y \mapsto 2]$, $u, v(d) \uparrow$, адже $u(d) \in A$, звідки $d \parallel_{-u, v} + u, v \mapsto u, v(d) = d \parallel_{-u, v} \neq d$.

Для ієрархічних даних можна переходити від послідовного до паралельного видалення та навпаки. Для випадку видалення порівнянних імен маємо ефект “поглинання”, див. твердження 1. В інших випадках маємо

Твердження 2. Нехай $u \div \{x_1, \dots, x_n\}$; тоді $d \parallel_{-u, x_1, \dots, x_n} = (d \parallel_{-u}) \parallel_{-x_1, \dots, x_n}$.

Зокрема, якщо $z \div u$, то $d \parallel_{-u, z} = (d \parallel_{-u}) \parallel_{-z} = (d \parallel_{-z}) \parallel_{-u}$.

Твердження 3. Для ієрархічних даних маємо $(z \mapsto d) \parallel_{-x_1, \dots, x_n} = z \mapsto d \parallel_{-z/x_1, \dots, z/x_n}$.

Приклад 6. Нехай u та z відрізняються префіксами – простими іменами; тоді $d \parallel_{-x, y, x, z} = d \parallel_{-x} + x \mapsto x(d) \parallel_{-y, z}$.

Твердження 4. Для ієрархічних даних маємо:

- 1) $x(d \parallel_{-z_1, \dots, z_n} + x \mapsto h) = h$;
- 2) $x(d \parallel_{-z_1, \dots, z_n}) = x(d)$, якщо $x \div \{z_1, \dots, z_n\}$;
- 3) $x(d \parallel_{-x, z_1, \dots, z_n}) \uparrow$.

Для кожного ієрархічного даного d можна подати $d \parallel_{-x_1, \dots, x_n}$ в певній стандартній формі, використовуючи тільки операції видалення компонент вигляду $\parallel_{-v_1, \dots, v_m}$ із простими іменами $v_1, \dots, v_m \in V$, а також операції $+$, іменування $v \mapsto$ та застосування d до імені, яке можна трактувати як розіменування $u_1(u_2(\dots u_k(d)\dots))$, де $u_1, \dots, u_k \in V$.

Для цього спочатку виділяємо в іменах x_1, \dots, x_n прості імена-префікси та розписуємо згідно тверджень 2, 4, 5, проводячи спрощення згідно твердження 1.

Наприклад, нехай $x_1 = v, u$. Тоді маємо $d \parallel_{-v, y, x_2, \dots, x_n} = (d \parallel_{-v, y}) \parallel_{-x_2, \dots, x_n} = (d \parallel_{-y} + v \mapsto v(d) \parallel_{-y}) \parallel_{-x_2, \dots, x_n} = d \parallel_{-v, x_2, \dots, x_n} + (v \mapsto v(d) \parallel_{-y}) \parallel_{-x_2, \dots, x_n} = d \parallel_{-v, x_2, \dots, x_n} + v \mapsto v(d) \parallel_{-y, v/x_2, \dots, v/x_n}$. Далі розписуємо аналогічно згідно виділених простих імен-префіксів для x_2, \dots, x_n , після чого поступово розписуємо операції видалення компонент в поіменованих цими префіксами даних, таких як отримане на першому кроці $v(d) \parallel_{-y, v/x_2, \dots, v/x_n}$, і т.д. Принагідно проводимо спрощення згідно твердження 1.

Приклад 7. $d \parallel_{-v, x, v, y, u, x, y} = (d \parallel_{-v} + v \mapsto v(d) \parallel_{-x}) \parallel_{-v, y, u, x, y} = d \parallel_{-v, v, y, u, x, y} + (v \mapsto v(d) \parallel_{-x}) \parallel_{-v, y} = d \parallel_{-v, u, x, y} + v \mapsto v(d) \parallel_{-x, y} = (d \parallel_{-u, x, y}) \parallel_{-v} + v \mapsto v(d) \parallel_{-x, y} = (d \parallel_{-u} + u \mapsto u(d) \parallel_{-x, y}) \parallel_{-v} + v \mapsto v(d) \parallel_{-x, y} = d \parallel_{-u, v} + (u \mapsto u(d) \parallel_{-x, y}) \parallel_{-v} + v \mapsto v(d) \parallel_{-x, y} = d \parallel_{-u, v} + u \mapsto (u(d) \parallel_{-x} + x \mapsto x(u(d)) \parallel_{-y}) \parallel_{-v} + v \mapsto v(d) \parallel_{-x, y} = d \parallel_{-u, v} + v \mapsto v(d) \parallel_{-x, y} + u \mapsto u(d) \parallel_{-x} + u \mapsto x \mapsto x(u(d)) \parallel_{-y}$.

Операцію ∇ накладення для випадку ієрархічних даних задамо так: $d \nabla h = h + d \parallel_{-m(h)}$.

Це означає, що для кожного термінального імені x ієрархічного даного h із d видаляємо всі компоненти вигляду $u \mapsto a$, де $a \in A$, $x \leq u$ чи $u \leq x$, після отримане ієрархічне дане об'єднуємо із h .

Задамо тепер на множині ієрархічних даних важливу операцію реномінації.

Параметризовану за іменами операцію реномінації $r_{x_1, \dots, x_n}^{v_1, \dots, v_n} : HD(V, A) \rightarrow HD(V, A)$ визначимо так:

$$r_{x_1, \dots, x_n}^{v_1, \dots, v_n}(d) = d \parallel_{-v_1, \dots, v_n} + v_1 \mapsto x_1(d) + \dots + v_n \mapsto x_n(d).$$

При цьому має виконуватися така умова коректності: усі імена v_1, \dots, v_n попарно не є порівнянними.

Операція реномінації монотонна в наступному смислі: якщо $d \subseteq h$, то $r_{x_1, \dots, x_n}^{v_1, \dots, v_n}(d) \subseteq r_{x_1, \dots, x_n}^{v_1, \dots, v_n}(h)$.

Приклад 8. Не завжди $r_{u, v}^{u, v}(d) = d : r_{u, v}^{u, v}([u \mapsto 1, x, y \mapsto 0]) = r_{u, v}^{u, v}([u \mapsto 1, x, y \mapsto 0]) = [u \mapsto 1, x, y \mapsto 0] \parallel_{-u, v} + u, v \mapsto u, v([u \mapsto 1, x, y \mapsto 0]) = [x, y \mapsto 0] + u, v \mapsto u, v([u \mapsto 1, x, y \mapsto 0]) = [x, y \mapsto 0]$, адже $u, v([u \mapsto 1, x, y \mapsto 0]) \uparrow$.

Використовуючи подання $d \parallel_{-x_1, \dots, x_n}$ в стандартній формі, операцію реномінації теж подамо в аналогічному стандартному вигляді, використовуючи операції $\parallel_{-v_1, \dots, v_m}$ із простими іменами $v_1, \dots, v_m \in V$, операції $+$, іменування $v \mapsto$ та потрактоване як розіменування $u_1(u_2(\dots u_k(d)\dots))$, де $u_1, \dots, u_k \in V$, а також застосування d до імені.

Приклад 9. Використовуючи (приклад 7) стандартне подання $d \parallel_{-v, x, v, y, u, x, y}$, отримуємо: $r_{u, y, v, x, v, x}^{v, x, v, y, u, x, y}(d) =$
 $= d \parallel_{-u, v} + v \mapsto v(d) \parallel_{-x, y} + u \mapsto u(d) \parallel_{-x} + u \mapsto x \mapsto x(u(d)) \parallel_{-y} + v \mapsto x \mapsto y(u(d)) + v \mapsto y \mapsto x(y(d)) + u \mapsto x \mapsto y \mapsto x(v(d)).$

Для логіки квазіарних предикатів, заданими над іменними множинами з простими іменами можна виконувати згортку композицій реномінації. Фактично це означає, що ми можемо описати застосування операції реномінації до даного, утвореного застосуванням операції реномінації, у вигляді однієї операції реномінації, $r_{x_1, \dots, x_n}^{v_1, \dots, v_n}(r_{y_1, \dots, y_n}^{u_1, \dots, u_m}(d)) = r_{x_1, \dots, x_n}^{v_1, \dots, v_n} \circ_{y_1, \dots, y_n}^{u_1, \dots, u_m}(d)$. Детальніше про згортку реномінацій див. [5]. Проте використання в реномінаціях складних імен робить в загальному випадку таке подання неможливим, про що легко пересвідчитись, розглядаючи приклад 9. Обійтися тут однією операцією вигляду $r_{x_1, \dots, x_n}^{v_1, \dots, v_n}$ без використання операцій $\parallel_{-y_1, \dots, y_m}$ неможливо, адже складні імена v_1, \dots, v_n та x_1, \dots, x_n можуть складатися тільки з простих імен x, u, v, z .

Для запису згортки реномінацій зі складними іменами будемо використовувати подання операції реномінації в стандартній формі.

Приклад 10. Маємо $r_{u, v}^z(r_x^u(d)) = r_{u, v}^z(d \parallel_{-u} + u \mapsto x(d)) = (d \parallel_{-u} + u \mapsto x(d)) \parallel_{-z} + z \mapsto v(u(d \parallel_{-u} + u \mapsto x(d))) =$
 $= d \parallel_{-u, z} + u \mapsto x(d) + z \mapsto v(x(d)) = r_{x, v, x}^{z, u}(d)$.

Приклад 11. $r_u^z(r_x^{u, v}(d)) = r_u^z(d \parallel_{-u, v} + u, v \mapsto x(d)) = r_u^z(d \parallel_{-u} + u \mapsto u(d) \parallel_{-v} + u, v \mapsto x(d)) = \varphi \parallel_{-z} + z \mapsto u(\varphi)$,
де $\varphi = d \parallel_{-u} + u \mapsto u(d) \parallel_{-v} + u, v \mapsto x(d)$.

Далі маємо $\varphi \parallel_{-z} + z \mapsto u(\varphi) = d \parallel_{-u, z} + u \mapsto v \mapsto x(d) + z \mapsto u(d) \parallel_{-v} + z \mapsto v \mapsto x(d)$. Подати цей вираз у вигляді однієї реномінації неможливо.

Приклад 12. Розпишемо $r_{x, x, y}^{u, v, z}(r_{z, y}^{x, v}(d))$: $r_{x, x, y}^{u, v, z}(r_{z, y}^{x, v}(d)) = r_{x, x, y}^{u, v, z}(d \parallel_{-x} + x \mapsto x(d) \parallel_{-y} + x \mapsto v \mapsto z, y(d)) =$
 $= \varphi \parallel_{-u, v, z} + u \mapsto v \mapsto x(\varphi) + z \mapsto x, y(\varphi)$, де $\varphi = d \parallel_{-x} + x \mapsto x(d) \parallel_{-y} + x \mapsto v \mapsto z, y(d)$. Тоді маємо $u(\varphi) = u(d)$,
 $y(x(\varphi)) = y(x(d))$. Стандартне подання: $r_{x, x, y}^{u, v, z}(r_{z, y}^{x, v}(d)) = \varphi \parallel_{-u, z} + u \mapsto u(\varphi) \parallel_{-z} + u \mapsto v \mapsto x(\varphi) + z \mapsto y(x(\varphi)) =$
 $= d \parallel_{-u, z, x} + u \mapsto u(d) \parallel_{-z} + u \mapsto v \mapsto x(d) \parallel_{-y} + u \mapsto v \mapsto v \mapsto y(z(d)) + z \mapsto y(x(\varphi)) + x \mapsto x(d) \parallel_{-y} + x \mapsto v \mapsto y(z(d)).$

3. Композиції предикатів над ієрархічними даними

Композиції визначають універсальні методи побудови предикатів, вони виступають ядром логіки певного типу. Згідно композиційно-номінативного підходу [2] композиції уточнюємо як функції (операції) над іменованими предикатами.

На **пропозиційному** рівні дані трактуються як абстрактні, тому предикати також трактуються як абстрактні, тобто як функції вигляду $P : D \rightarrow \{T, F\}$, де D – абстрактна множина.

Композиції пропозиційного рівня називають логічними зв'язками. Основними логічними зв'язками є 1-арна композиція заперечення \neg та бінарні диз'юнкція \vee , кон'юнкція $\&$, імплікація \rightarrow , еквіваленція \leftrightarrow .

Композиції \neg та \vee називають базовими пропозиційними композиціями.

На **номінативних** рівнях дані будуються із множин предметних імен та предметних значень. Такі дані називають номінативними. Функції та предикати, задані на номінативних даних, називають квазіарними.

Логіки квазіарних предикатів, заданих на іменних множинах – однорівневих номінативних даних – детально досліджені в [5]. Для таких логік були виділені наступні підрівні: реномінативний, кванторний, кванторно-екваційний, функціональний та функціонально-екваційний.

У даній роботі досліджуються логіки часткових предикатів, заданих на описаних в попередньому розділі ієрархічних номінативних даних. Такі предикати теж називатимемо квазіарними.

Функцію вигляду $P : HD(V, A) \rightarrow \{T, F\}$ назвемо квазіарним предикатом на $HD(V, A)$.

Квазіарні предикати на $HD(V, A)$ будемо також називати (V, A) -квазіарними.

Множину квазіарних предикатів на $HD(V, A)$ позначимо $Pr^{V, A}$.

Ім'я $x \in V$ строго неістотне для квазіарного предиката P на $HD(V, A)$, якщо для довільних $d, \alpha \in HD(V, A)$ маємо $P(d \parallel_{-x} + x \mapsto \alpha) = P(d \parallel_{-x})$.

V -квазіарний предикат $P : HD(V, A) \rightarrow \{T, F\}$ назвемо еквітонним, якщо для довільних $d, d' \in HD(V, A)$ із умови $d \subseteq d'$ та $P(d) \downarrow$ випливає $P(d') \downarrow = P(d)$.

Будемо тут розглядати логіки квазіарних предикатів над ієрархічними номінативними даними на реномінативному та кванторному рівнях.

На **реномінативному** рівні базовими композиціями є \vee, \neg та реномінація $R_{x_1, \dots, x_n}^{v_1, \dots, v_n}$.

Відмінність композиції реномінації $R_{x_1, \dots, x_n}^{v_1, \dots, v_n}$ для предикатів над ієрархічними номінативними даними від традиційної композиції реномінації [5] для предикатів над однорівневими номінативними даними (іменними множинами) полягає найперше у тому, що імена v_1, \dots, v_n та x_1, \dots, x_n можуть бути складними. Як і в традиційному випадку [5], композицію реномінації визначаємо через операцію реномінації на даних.

1-арна параметрична композиція реномінації $R_{x_1, \dots, x_n}^{v_1, \dots, v_n}$ кожному (V, A) -квазіарному предикату Q зіставляє (V, A) -квазіарний предикат $R_{x_1, \dots, x_n}^{v_1, \dots, v_n}(Q)$, значення якого для кожного $d \in HD(V, A)$ задається так:

$$R_{x_1, \dots, x_n}^{v_1, \dots, v_n}(Q)(d) = Q(r_{x_1, \dots, x_n}^{v_1, \dots, v_n}(d)).$$

Зрозуміло, що тоді $R_{x_1, \dots, x_n}^{v_1, \dots, v_n}(Q)(d) = Q(d \parallel_{-v_1, \dots, v_n} + v_1 \mapsto x_1(d) + \dots + v_n \mapsto x_n(d))$.

Введемо позначення вигляду \bar{y} для y_1, \dots, y_n . Тоді замість $r_{x_1, \dots, x_n}^{v_1, \dots, v_n}$ та $R_{x_1, \dots, x_n}^{v_1, \dots, v_n}$ будемо писати $r_{\bar{x}}^{\bar{y}}$ та $R_{\bar{x}}^{\bar{y}}$.

Розглянемо основні властивості композицій $R_{\bar{x}}^{\bar{y}}$.

$$R \neg) R_{\bar{x}}^{\bar{y}}(\neg P) = \neg R_{\bar{x}}^{\bar{y}}(P);$$

$$R \vee) R_{\bar{x}}^{\bar{y}}(P \vee Q) = R_{\bar{x}}^{\bar{y}}(P) \vee R_{\bar{x}}^{\bar{y}}(Q).$$

Аналогічно записуються властивості $R \rightarrow$, $R \&$, $R \leftrightarrow$.

$$RR) R_{\bar{x}}^{\bar{y}}(R_{\bar{y}}^{\bar{u}}(P)(d) = P(r_{\bar{y}}^{\bar{u}}(r_{\bar{x}}^{\bar{y}}(d))) \text{ для кожного } d \in HD(V, A).$$

RSN) Нехай ім'я $y \in V$ строго неістотне для предиката P ; тоді $R_{z, \bar{x}}^{y, \bar{y}}(P) = R_{\bar{x}}^{\bar{y}}(P)$.

$$RT) R_{z, \bar{x}}^{z, \bar{y}}(P) = R_{\bar{x}}^{\bar{y}}(P) \text{ при умові } z \in V.$$

Враховуючи п. 5 твердження 1, у випадку еквітонних предикатів для складних імен маємо:

$$RTE) R_{u, \bar{x}}^{u, \bar{y}}(P) \cong R_{\bar{x}}^{\bar{y}}(P) \text{ (тут } \cong \text{ означає рівність з точністю до визначеності, див. [5]).}$$

На **кванторному** рівні базовими композиціями є \vee , \neg , $R_{x_1, \dots, x_n}^{v_1, \dots, v_n}$, $\exists x$.

Дамо визначення 1-арних параметричних композицій квантифікації $\exists x$ та $\forall x$. На відміну від традиційного випадку [5], кванторне ім'я x може бути складним, крім того, квантифікацію можна вести як за всіма ієрархічними даними, так і за базовими даними. В даній роботі розглядаємо тільки квантифікацію за базовими даними.

Предикати $\exists x(P)$ та $\forall x(P)$ будемо позначати $\exists xP$ та $\forall xP$. Зазначені предикати задамо так:

$$\exists xP(d) = \begin{cases} T, & \text{якщо існує } b \in A: P(d \parallel_{-x} + x \mapsto b) = T, \\ F, & \text{якщо } P(d \parallel_{-x} + x \mapsto a) \downarrow = F \text{ для всіх } a \in A, \\ & \text{невизначене в усіх інших випадках.} \end{cases}$$

$$\forall xP(d) = \begin{cases} F, & \text{якщо існує } b \in A: P(d \parallel_{-x} + x \mapsto b) = F, \\ T, & \text{якщо } P(d \parallel_{-x} + x \mapsto a) \downarrow = T \text{ для всіх } a \in A, \\ & \text{невизначене в усіх інших випадках.} \end{cases}$$

Наведемо основні властивості композицій $\exists x$ та $\forall x$.

$$1. \text{ Якщо } x \text{ та } y \text{ непорівнянні, то } \exists x \exists y P = \exists y \exists x P \text{ та } \forall x \forall y P = \forall y \forall x P.$$

$$2. \neg \exists x P = \forall x \neg P; \neg \forall x P = \exists x \neg P.$$

$$3. \exists x P \vee \exists x Q = \exists x (P \vee Q); \forall x P \& \forall x Q = \forall x (P \& Q).$$

4. Поглинання зовнішнього квантора внутрішнім за тим же іменем:

$$\exists x \exists x P = \exists x P; \forall x \exists x P = \exists x P; \exists x \forall x P = \forall x P; \forall x \forall x P = \forall x P.$$

Вищенаведені властивості справджуються як для простих, так і для складних імен.

Наведемо специфічні властивості поглинання для складних порівнянних імен.

5. Поглинання зовнішнього квантора внутрішнім за загальнішим іменем:

$$\exists x. y \exists x P = \exists x P; \forall x. y \exists x P = \exists x P; \exists x. y \forall x P = \forall x P; \forall x. y \forall x P = \forall x P.$$

$$6. \exists x \exists x. y P = \forall x \exists x. y P \neq \exists x P; \exists x \exists x. y P \neq \exists x. y P; \exists x \forall x. y P = \forall x \forall x. y P \neq \forall x P; \exists x \forall x. y P \neq \forall x. y P.$$

Залучаючи до розгляду композицію реномінації, отримуємо такі закони.

$$7. \exists z (R_y^x P) = R_y^x (\exists z P), \text{ якщо } z \div y \text{ та } z \div x.$$

$$\text{Узагальнення: } \exists z (R_{\bar{y}}^{\bar{x}} P) = R_{\bar{y}}^{\bar{x}} (\exists z P), \text{ якщо } z \div \{\bar{u}, \bar{v}\}.$$

8. Поглинання кванторного імені загальнішим верхнім іменем реномінації:

$$\exists x. y (R_z^x P) = R_z^x (P), \text{ якщо } x. y \div z.$$

$$\text{Водночас } \exists x. y (R_z^x P) \neq R_z^x (\exists x. y P) \text{ та } \exists x. y (R_{x. y. v}^x P) \neq R_{x. y. v}^x (P).$$

$$\text{Узагальнення: } \exists x. y (R_{z, \bar{v}}^{x, \bar{u}} P) = R_{z, \bar{v}}^{x, \bar{u}} (P), \text{ якщо } x. y \div \{z, \bar{u}, \bar{v}\}.$$

9. Поглинання верхнього імені реномінації загальнішим кванторним іменем: $R_u^{x,y}(\exists xP) = \exists xP$.

Водночас $R_u^{x,y}(\exists xP) \neq \exists x(R_u^{x,y}(P))$, $R_u^{x,y}(\exists xP) \neq R_u^{x,y}(P)$, $\exists x(R_{x,u}^z P) \neq R_{x,u}^z(\exists xP)$.

Узагальнення:

$$R_{z,\bar{v}}^{\bar{y}}(\exists xP) = R_{\bar{v}}^{\bar{y}}(\exists xP), \text{ якщо } x \text{ є префіксом імен із } \bar{y} \text{ та } x \div \{\bar{u}\}.$$

Властивості 7–9 відповідним чином переформулюються для $\forall z$.

4. Семантичні моделі та мови логік квазіарних предикатів над ієрархічними даними

Семантичними моделями композиційно-номінативних логік над ієрархічними номінативними даними (скорочено КНЛІ) є композиційні системи квазіарних предикатів вигляду $(HD(V,A), Pr^{V,A}, C)$.

Для КНЛІ кванторного рівня множина C задається множиною базових композицій $\{\neg, \vee, R_{\bar{x}}^{\bar{v}}, \exists x\}$, для КНЛІ реномінативного рівня – множиною $\{\neg, \vee, R_{\bar{x}}^{\bar{v}}\}$.

При зафіксованій C система $(HD(V,A), Pr^{V,A}, C)$ однозначно визначається об'єктом вигляду $(HD(V,A), Pr^{V,A})$ – алгебраїчною системою (АС) з квазіарними предикатами над ієрархічними номінативними даними. Такі алгебраїчні системи будемо скорочено позначати $(A, Pr^{V,A})$.

Розгляд композиційних систем передбачає наявність мови логіки, індукованої відповідними інтенціональними моделями (рівнями розгляду). Це означає необхідність позначення базових предикатів, із яких за допомогою композицій будуються складніші предикати.

Алфавіт мови КНЛІ кванторного рівня: символи базових композицій, множина Ps предикатних символів (сигнатура мови), множина V базових предметних імен.

Дамо визначення множини Fr формул мови КНЛІ кванторного рівня:

- 1) кожний предикатний символ (ПС) є формулою; такі формули атомарні;
- 2) якщо Φ та Ψ – формули, то $\neg\Phi$ та $\vee\Phi\Psi$ – формули;
- 3) якщо Φ – формула, то $R_{\bar{x}}^{\bar{v}}\Phi$ – формула;
- 4) якщо Φ – формула, то $\exists x\Phi$ – формула.

Для випадку КНЛІ реномінативного рівня у цьому визначенні опускаємо п. 4.

Позначимо $nm(\Phi)$ множину всіх імен, які фігурують у символах реномінації та квантифікації формули Φ , $snm(\Phi)$ – множину всіх простих імен, які фігурують у символах реномінації та квантифікації формули Φ .

Щоб відрізнити символи базових композицій як елементів алфавіту мови від позначень цих композицій у виразах, які подають складніші предикати через простіші, останні позначення записуватимемо жирним текстом.

Конкретна інтерпретація мови визначається АС $(A, Pr^{V,A})$ та значеннями ПС на $HD(V,A)$, які задаємо за допомогою тотального однозначного відображення $I: Ps \rightarrow Pr^{V,A}$. Тому моделями мови КНЛІ будемо вважати об'єкти вигляду $((A, Pr^{V,A}), I)$ – АС із квазіарними предикатами над ієрархічними номінативними даними з доданою сигнатурою. Будемо їх скорочено позначати у вигляді (A, I) .

Відображення інтерпретації формул $J: Fr \rightarrow Pr^{V,A}$ визначається за допомогою I так:

- 1) $J(p) = I(p)$ для кожного $p \in Ps$;
- 2) $J(\neg\Phi) = \neg(J(\Phi))$, $J(\vee\Phi\Psi) = \vee(J(\Phi), J(\Psi))$;
- 3) $J(R_{\bar{x}}^{\bar{v}}\Phi) = R_{\bar{x}}^{\bar{v}}(J(\Phi))$;
- 4) $J(\exists x\Phi) = \exists x(J(\Phi))$.

Для випадку КНЛІ реномінативного рівня опускаємо п. 4.

Предикат $J(\Phi)$ – значення формули Φ при інтерпретації на $A = (A, I)$, – позначаємо Φ_A .

Формула Φ частково істинна при інтерпретації на $A = (A, I)$, або A -неспростовна, якщо Φ_A – частково істинний (неспростовний) предикат. Це позначаємо $A \models \Phi$.

Формула Φ всюди (частково) істинна, або неспростовна, якщо Φ частково істинна при кожній інтерпретації. Це позначаємо $\models \Phi$.

На множині формул мови КНЛІ введемо відношення логічного наслідку \models слабкого логічного наслідку \models та логічної еквівалентності \sim .

Формула Ψ є логічним наслідком формули Φ , що позначимо $\Phi \models \Psi$, якщо формула $\Phi \rightarrow \Psi$ неспростовна.

Формула Ψ є слабким логічним наслідком формули Φ , що позначимо $\Phi \models \Psi$, якщо для кожної $A = (A, I)$ із умови $A \models \Phi$ випливає $A \models \Psi$.

Формули Φ та Ψ логічно еквівалентні, що позначимо $\Phi \sim \Psi$, якщо $\Phi \models \Psi$ та $\Psi \models \Phi$.

Φ та Ψ логічно еквівалентні (позначаємо $\Phi \sim \Psi$), якщо $\Phi \models \Psi$ та $\Psi \models \Phi$.

Φ та Ψ логічно строго еквівалентні [7] (позначаємо $\Phi \sim_{TF} \Psi$), якщо $T(\Phi_A) = T(\Psi_A)$ та $F(\Psi_A) = F(\Phi_A)$ для кожної АС A . $\Phi \sim_{TF} \Psi$ означає, що Φ та Ψ завжди інтерпретуються як один і той же предикат.

Розглянемо наслідки з порожньої множини формул.

$\emptyset \models \Phi$ означає, що $F(\Phi_A) = \emptyset$ для кожної АС A , тобто Φ всюди частково істинна, або неспростовна.

Поширимо відношення логічного наслідку на довільні множини формул.

Нехай Γ та Δ – множини формул мови сигнатури Ps . Нехай $A = (A, I)$ – АС сигнатури Ps .

Скажемо, що Δ є логічним наслідком Γ в АС A , якщо для всіх $d \in HD(V, A)$ із того, що $\Phi_A(d) = T$ для всіх $\Phi \in \Gamma$, випливає, що неможливо $\Psi_A(d) = F$ для всіх $\Psi \in \Delta$. Цей факт позначаємо $\Gamma_A \models \Delta$.

Скажемо, що Δ є логічним наслідком Γ , якщо $\Gamma_A \models \Delta$ для всіх АС $A = (A, I)$ сигнатури Ps .

Те, що Δ є логічним наслідком Γ , позначаємо $\Gamma \models \Delta$.

Як і для випадку композиційно-номінативних логік над однорівневими іменними множинами [5], відношення логічного наслідку для множин формул рефлексивне, але нетранзитивне.

Для КНЛІ справджуються теореми семантичної еквівалентності та заміни еквівалентних.

Теорема 1 (семантичної еквівалентності). Нехай Φ' отримано з формули Φ заміною деяких входжень формул Φ_1, \dots, Φ_n на Ψ_1, \dots, Ψ_n відповідно. Якщо $\Phi_1 \sim \Psi_1, \dots, \Phi_n \sim \Psi_n$, то $\Phi \sim \Phi'$.

Теорема 2 (семантичної еквівалентності, сильна форма). Нехай Φ' отримано з формули Φ заміною деяких входжень формул Φ_1, \dots, Φ_n на Ψ_1, \dots, Ψ_n відповідно. Якщо $\Phi_1 \sim_{TF} \Psi_1, \dots, \Phi_n \sim_{TF} \Psi_n$, то $\Phi \sim_{TF} \Phi'$.

Доведення теорем еквівалентності – індукцією за побудовою формули.

Теорема 3 (заміни еквівалентних). Нехай $\Phi \sim \Psi$. Тоді $\Phi, \Gamma \models \Delta \Leftrightarrow \Psi, \Gamma \models \Delta$ та $\Gamma \models \Delta, \Phi \Leftrightarrow \Gamma \models \Delta, \Psi$.

Ім'я $x \in V$ строго неістотне для формули Φ , якщо при інтерпретації на кожній $A = (A, I)$ і м'я x строго неістотне для предиката Φ_A .

Твердження 5. Нехай $y \in V$ строго неістотне для формули Φ . Тоді $\exists x \Phi \sim_{TF} \exists y R_y^x \Phi$.

Для кожного $p \in Ps$ множину строго неістотних предметних імен задамо за допомогою тотальної функції $v: Ps \rightarrow 2^V$. Для КНЛІ постулюємо нескінченність множини $V_{Ts} = \bigcap_{p \in Ps} v(p)$ тотально строго неістотних імен.

Наведемо властивості формул, пов'язані з композицією реномінації. Зазначені властивості записуються згідно відповідних властивостей композиції реномінації для квазіарних предикатів над ієрархічними номінативними даними і мають таку ж назву. Властивості, не пов'язані з кванторами, справджуються вже для КНЛІ реномінативного рівня. При кожній інтерпретації предикати, що є значеннями виділених формул, збігаються, тобто в цих властивостях маємо відношення строгої еквівалентності.

RSN) Нехай ім'я $y \in V$ строго неістотне для Φ . Тоді $R_{z,\bar{x}}^{y,\bar{v}}(\Phi) \sqsubseteq_{TF} R_{\bar{x}}^{\bar{v}}(\Phi)$.

RT) $R_{z,\bar{x}}^{z,\bar{v}}(\Phi) \sqsubseteq_{TF} R_{\bar{x}}^{\bar{v}}(\Phi)$ при умові $z \in V$; зокрема, $R_z^z(\Phi) \sqsubseteq_{TF} \Phi$.

R¬) $R_{\bar{x}}^{\bar{v}}(\neg\Phi) \sqsubseteq_{TF} \neg R_{\bar{x}}^{\bar{v}}(\Phi)$.

R∨) $R_{\bar{x}}^{\bar{v}}(\Phi \vee \Psi) \sqsubseteq_{TF} R_{\bar{x}}^{\bar{v}}(\Phi) \vee R_{\bar{x}}^{\bar{v}}(\Psi)$.

Узагальнюючи R¬ та R∨, отримуємо RR¬ та RR∨.

RR¬) $R_{\bar{x}}^{\bar{v}}(R_{\bar{y}}^{\bar{w}}(\dots R_{\bar{z}}^{\bar{u}}(\neg\Phi)\dots)) \sqsubseteq_{TF} \neg R_{\bar{x}}^{\bar{v}}(R_{\bar{y}}^{\bar{w}}(\dots R_{\bar{z}}^{\bar{u}}(\Phi)\dots))$.

RR∨) $R_{\bar{x}}^{\bar{v}}(R_{\bar{y}}^{\bar{w}}(\dots R_{\bar{z}}^{\bar{u}}(\Phi \vee \Psi)\dots)) \sqsubseteq_{TF} R_{\bar{x}}^{\bar{v}}(R_{\bar{y}}^{\bar{w}}(\dots R_{\bar{z}}^{\bar{u}}(\Phi)\dots)) \vee R_{\bar{x}}^{\bar{v}}(R_{\bar{y}}^{\bar{w}}(\dots R_{\bar{z}}^{\bar{u}}(\Psi)\dots))$.

Подібним чином можна записати властивості R&, R→, R↔, RR&, RR→, RR↔.

Записати властивість RR для згортки реномінацій в аналогічному вигляді, із використанням \sim_{TF} , вже неможливо. Можна лише переписати властивість RR композиції реномінації:

RR_C) для кожних $A = (A, I)$, $d \in HD(V, A)$ маємо $R_{\bar{x}}^{\bar{v}}(R_{\bar{y}}^{\bar{w}}(\Phi_A)(d) = \Phi_A(r_{\bar{y}}^{\bar{w}}(r_{\bar{x}}^{\bar{v}}(d))))$.

Залучаючи до реномінації квантори, отримуємо такі властивості:

ANQ) $\exists x.y(R_{z,\bar{v}}^{x,\bar{u}}\Phi) \sqsubseteq_{TF} R_{z,\bar{v}}^{x,\bar{u}}(\Phi)$ та $\forall x.y(R_{z,\bar{v}}^{x,\bar{u}}\Phi) \sqsubseteq_{TF} R_{z,\bar{v}}^{x,\bar{u}}(\Phi)$, якщо $x.y \div \{z, \bar{u}, \bar{v}\}$.

Зокрема, $\exists x.y(R_z^x\Phi) \sqsubseteq_{TF} R_z^x(\Phi)$ та $\forall x.y(R_z^x\Phi) \sqsubseteq_{TF} R_z^x(\Phi)$, якщо $x.y \div z$.

ANR) $R_{z,\bar{v}}^{\bar{y},\bar{u}}(\exists x\Phi) \sqsubseteq_{TF} R_{\bar{v}}^{\bar{u}}(\exists x\Phi)$ та $R_{z,\bar{v}}^{\bar{y},\bar{u}}(\forall x\Phi) \sqsubseteq_{TF} R_{\bar{v}}^{\bar{u}}(\forall x\Phi)$, якщо x є префіксом імен із \bar{y} та x непорівнянне із кожним з імен \bar{u} . Зокрема, $R_u^{x,y}(\exists x\Phi) \sqsubseteq_{TF} \exists x\Phi$ та $R_u^{x,y}(\forall x\Phi) \sqsubseteq_{TF} \forall x\Phi$.

R∃) $R_{\bar{v}}^{\bar{u}}(\exists y\Phi) \sqsubseteq_{TF} \exists y(R_{\bar{v}}^{\bar{u}}\Phi)$, якщо $y \div \{\bar{u}, \bar{v}\}$.

R∃∃) $R_{\bar{x}}^{\bar{v}}(\exists y\Phi) \sqsubseteq_{TF} \exists z R_{\bar{x}}^{\bar{v}}(R_z^y(\Phi))$ за умови, що z тотально строго неістотне та $z \notin \text{snm}(R_{\bar{x}}^{\bar{v}}(\exists y\Phi))$.

Властивість R∃∃ використовуємо, якщо у порівнянне хоча б із одним з імен \bar{u} та \bar{v} .

Подібним чином можна записати R∨ та R∨∨. Властивості R∃, R∃∃, R∨, R∨∨ узагальнюються до RR∃, RR∃∃, RR∨, RR∨∨ аналогічно тому, як R¬ та R∨ узагальнюються до RR¬ та RR∨.

Для логік еквітонних предикатів RT узагальнюється із певним ослабленням (\sim замість \sim_{TF}) до RTE:

RTE) $R_{u,\bar{x}}^{\bar{y},\bar{v}}(\Phi) \sqsubseteq R_{\bar{x}}^{\bar{v}}(\Phi)$; зокрема, $R_u^u(\Phi) \sqsubseteq \Phi$.

Формулу вигляду $R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(p)\dots))$, де $p \in Ps$ та в реномінаціях згорнуті тотожні пари простих імен, назовемо слабо примітивною. Якщо при цьому в реномінаціях згорнуті тотожні пари як простих, так і складних імен, таку формулу назовемо примітивною.

Для логік еквітонних предикатів будемо розглядати примітивні формули.

Із кожною примітивною формулою $R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(p)\dots))$ зв'яжемо вираз вигляду $p(\alpha)$, де α – подана в стандартній формі згортка реномінацій $r_{\bar{x}}^{\bar{u}}(r_{\bar{y}}^{\bar{v}}(\dots r_{\bar{z}}^{\bar{w}}(\delta)\dots))$, δ – символ, який позначає довільне дане, $\delta \notin V \cup Ps$.

Такий вираз назовемо субреномінантою примітивної формули $R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(p)\dots))$.

Для врахування строго неістотності предметних імен у стандартній формі α згортки реномінацій видаємо усі компоненти зі старшими простими іменами z такими, що $z \in v(p)$. Тоді вираз $p(\alpha)$ назовемо реномінантою примітивної формули $R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(p)\dots))$.

Наведемо тепер основні властивості композицій квантифікації.

Q1. $\exists x \exists y \Phi \sim_{TF} \exists y \exists x \Phi$ та $\forall x \forall y \Phi \sim_{TF} \forall y \forall x \Phi$, якщо x та y непорівнянні.

Q2. $\neg \forall x \Phi \sim_{TF} \exists x \neg \Phi$ та $\neg \exists x \Phi \sim_{TF} \forall x \neg \Phi$.

Q3. $\exists x \Phi \sim_{TF} \forall x \exists x \Phi$, $\exists x \Phi \sim_{TF} \exists x \exists x \Phi$; $\forall x \Phi \sim_{TF} \forall x \forall x \Phi$, $\forall x \Phi \sim_{TF} \exists x \forall x \Phi$.

Q4. $\exists x \Phi \sim_{TF} \forall x. y \exists x \Phi$, $\exists x \Phi \sim_{TF} \exists x. y \exists x \Phi$; $\forall x \Phi \sim_{TF} \forall x. y \forall x \Phi$, $\forall x \Phi \sim_{TF} \exists x. y \forall x \Phi$.

Q5. $\exists x \exists x. y \Phi \sim_{TF} \forall x \exists x. y \Phi$, $\exists x \forall x. y \Phi \sim_{TF} \forall x \forall x. y \Phi$.

Q6. $\exists x \Phi \vee \exists x \Psi \sim_{TF} \exists x (\Phi \vee \Psi)$ та $\forall x \Phi \& \forall x \Psi \sim_{TF} \forall x (\Phi \& \Psi)$.

На додаток до властивостей, які описують строго еквівалентність формул, наведемо також властивості, пов'язані з відношенням логічного наслідку; вони аналогічні відповідним властивостям логіки квазіарних предикатів над однорівневими іменними множинами.

Q7. $\exists x (\Phi \& \Psi) \models \exists x \Phi \& \exists x \Psi$ та $\forall x \Phi \vee \forall x \Psi \models \forall x (\Phi \vee \Psi)$.

Q8. $\exists y \forall x \Phi \models \forall x \exists y \Phi$; водночас не завжди $\forall x \exists y \Phi \models \exists y \forall x \Phi$.

Q9. $\Phi \models \exists x \Phi$ та $\Phi \models \forall x \Phi$.

Q10. $\models \exists y \forall x \Phi \rightarrow \forall x \exists y \Phi$, але не завжди $\models \forall x \exists y \Phi \rightarrow \exists y \forall x \Phi$.

Q11. $\models \forall x (\forall x \Phi \rightarrow \Phi)$ та $\models \exists x (\forall x \Phi \rightarrow \Phi)$; $\models \forall x (\Phi \rightarrow \exists x \Phi)$ та $\models \exists x (\Phi \rightarrow \exists x \Phi)$.

Розглянемо властивості відношення логічного наслідку \models для множин формул.

Властивості відношення \models на пропозиційному рівні ідентичні відповідним властивостям логіки квазіарних предикатів над однорівневими іменними множинами (див. [5]).

Розглянемо властивості відношення логічного наслідку для множин формул, пов'язані з композицією реномінації. Не пов'язані з кванторами властивості справджуються вже для логік реномінативного рівня. При інтерпретації на кожній $A = (A, I)$, предикати, що є значеннями виділених формул, збігаються (за винятком RTE).

RTE $_{\perp}$) $R_{u,x}^{u,\bar{v}}(\Phi), \Gamma_A \models \Delta \Leftrightarrow R_{\bar{x}}^{\bar{v}}(\Phi), \Gamma_A \models \Delta$.

RTE $_{\downarrow}$) $\Gamma_A \models \Delta, R_{u,x}^{u,\bar{v}}(\Phi) \Leftrightarrow \Gamma_A \models \Delta, R_{\bar{x}}^{\bar{v}}(\Phi)$.

RsN $_{\perp}$) $R_{z,x}^{y,\bar{v}}(\Phi), \Gamma_A \models \Delta \Leftrightarrow R_{\bar{x}}^{\bar{v}}(\Phi), \Gamma_A \models \Delta$ за умови, що $y \in V$ строго неістотне для Φ .

RsN $_{\downarrow}$) $\Gamma_A \models \Delta, R_{z,x}^{y,\bar{v}}(\Phi) \Leftrightarrow \Gamma_A \models \Delta, R_{\bar{x}}^{\bar{v}}(\Phi)$ за умови, що $y \in V$ строго неістотне для Φ .

RR $_{\perp}$) $R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(\neg \Phi)\dots)), \Gamma_A \models \Delta \Leftrightarrow \neg R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(\Phi)\dots)), \Gamma_A \models \Delta$.

RR $_{\downarrow}$) $\Gamma, R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(\neg \Phi)\dots))_A \models \Delta \Leftrightarrow \Gamma, \neg R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(\Phi)\dots))_A \models \Delta$.

RR $_{\vee}$) $R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(\Phi \vee \Psi)\dots)), \Gamma_A \models \Delta \Leftrightarrow R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(\Phi)\dots)) \vee R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(\Psi)\dots)), \Gamma_A \models \Delta$.

RR $_{\vee}$) $\Gamma, R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(\Phi \vee \Psi)\dots))_A \models \Delta \Leftrightarrow \Gamma, R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(\Phi)\dots)) \vee R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(\Psi)\dots))_A \models \Delta$.

Аналогічно записуються властивості RR $_{\rightarrow}$, RR $_{\rightarrow}$, RR $_{\&}$, RR $_{\&}$.

R $_{\exists}$) $R_{\bar{x}}^{\bar{u}}(\exists y \Phi), \Gamma_A \models \Delta \Leftrightarrow \exists y R_{\bar{x}}^{\bar{u}}(\Phi), \Gamma_A \models \Delta$ за умови, що $y \notin \{\bar{u}, \bar{v}\}$.

R $_{\exists}$) $\Gamma_A \models \Delta, R_{\bar{x}}^{\bar{u}}(\exists y \Phi) \Leftrightarrow \Gamma_A \models \Delta, \exists y R_{\bar{x}}^{\bar{u}}(\Phi)$ за умови, що $y \notin \{\bar{u}, \bar{v}\}$.

R $_{\exists}$) $R_{\bar{x}}^{\bar{u}}(\exists y \Phi), \Gamma_A \models \Delta \Leftrightarrow \exists z R_{\bar{x}}^{\bar{u}}(R_{\bar{z}}^{\bar{v}}(\Phi)), \Gamma_A \models \Delta$.

R $_{\exists}$) $\Gamma_A \models \Delta, R_{\bar{x}}^{\bar{u}}(\exists y \Phi) \Leftrightarrow \Gamma_A \models \Delta, \exists z R_{\bar{x}}^{\bar{u}}(R_{\bar{z}}^{\bar{v}}(\Phi))$.

Для R $_{\exists}$ та R $_{\exists}$ умова: z тотально строго неістотне та $z \notin snm(R_{\bar{x}}^{\bar{u}}(\exists y \Phi))$.

Аналогічно записуються властивості R $_{\forall}$, R $_{\forall}$, R $_{\forall}$, R $_{\forall}$.

Властивості типу R $_{\exists}$, R $_{\exists}$, R $_{\forall}$, R $_{\forall}$ узагальнюються до властивостей типу RR $_{\exists}$, RR $_{\exists}$, RR $_{\forall}$, RR $_{\forall}$.

5. Секвенційні числення логік квазіарних предикатів над ієрархічними даними

Для логік еквітонних квазіарних предикатів над ієрархічними даними побудуємо числення секвенційного типу. Обмежимося тут розглядом секвенційних числень для логік реномінативного рівня. Секвенції трактуємо як множини специфікованих формул. Кожна формула секвенції відмічена (специфікована) зліва одним з двох символів \vdash чи \dashv . Секвенції позначаємо у вигляді $\vdash \Gamma \dashv \Delta$, де усі формули множини Γ специфіковані \vdash , усі формули множини Δ – символом \dashv . Не деталізуючи, секвенції позначатимемо Σ .

Секвенційне числення будується так, що секвенція $\vdash \Gamma \dashv \Delta$ має виведення тоді й тільки тоді, коли $\Gamma \models \Delta$.

Секвенція Σ замкнена, якщо існує Φ така, що $\vdash \Phi \in \Sigma$ та $\dashv \Phi \in \Sigma$, або якщо існують примітивні ϕ та ψ з однако-вими реномінантами такі, що $\vdash \phi \in \Sigma$ та $\dashv \psi \in \Sigma$. Отже, якщо $\vdash \Gamma \dashv \Delta$ замкнена, то $\Gamma \models \Delta$.

Виведення в секвенційних численнях має вигляд дерева, вершинами якого є секвенції. Такі дерева називають секвенційними (див. [5]). Секвенційне дерево замкнене, якщо кожний його листок – замкнена секвенція.

Секвенція Σ вивідна, або має виведення, якщо існує замкнене секвенційне дерево з коренем Σ . Таке дере-во назвемо виведенням секвенції Σ .

Семантичним властивостям відношення \models зіставимо їх синтаксичні аналоги – секвенційні форми.

Наведемо секвенційні форми для реномінативних логік еквітонних квазіарних предикатів над ієрархіч-ними даними.

$$\begin{array}{l} \vdash \neg \frac{\dashv A, \Sigma}{\vdash \neg A, \Sigma} \qquad \dashv \neg \frac{\vdash A, \Sigma}{\dashv \neg A, \Sigma} \\ \vdash \vee \frac{\vdash A, \Sigma \quad \vdash B, \Sigma}{\vdash A \vee B, \Sigma} \qquad \dashv \vee \frac{\dashv A, \dashv B, \Sigma}{\dashv A \vee B, \Sigma} \\ \vdash \text{RTE} \frac{\vdash R_{\bar{x}}^{\bar{v}}(A), \Sigma}{\vdash R_{u, \bar{x}}^{u, \bar{v}}(A), \Sigma} \qquad \dashv \text{RTE} \frac{\dashv R_{\bar{x}}^{\bar{v}}(A), \Sigma}{\dashv R_{u, \bar{x}}^{u, \bar{v}}(A), \Sigma} \\ \vdash \text{RR}\neg \frac{\vdash \neg R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(A)\dots)), \Sigma}{\vdash R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(\neg A)\dots)), \Sigma} \qquad \dashv \text{RR}\neg \frac{\dashv \neg R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(A)\dots)), \Sigma}{\dashv R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(\neg A)\dots)), \Sigma} \\ \vdash \text{RR}\vee \frac{\vdash R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(A)\dots)) \vee R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(B)\dots)), \Sigma}{\vdash R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(A \vee B)\dots)), \Sigma} \qquad \dashv \text{RR}\vee \frac{\dashv R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(A)\dots)) \vee R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(B)\dots)), \Sigma}{\dashv R_{\bar{x}}^{\bar{u}}(R_{\bar{y}}^{\bar{v}}(\dots R_{\bar{z}}^{\bar{w}}(A \vee B)\dots)), \Sigma} \end{array}$$

Секвенційні числення із вищенаведеними базовими секвенційними формами назвемо *RID*-численнями.

Для побудованих секвенційних *RID*-числень справджуються теореми коректності та повноти.

Теорема 4 (коректності). Нехай секвенція $\vdash \Gamma \dashv \Delta$ вивідна. Тоді $\Gamma \models \Delta$.

Доведення теореми проводиться індукцією за побудовою замкненого секвенційного дерева для $\vdash \Gamma \dashv \Delta$.

Теорема 5 (повноти). Нехай $\Gamma \models \Delta$. Тоді секвенція $\vdash \Gamma \dashv \Delta$ вивідна.

Для доведення повноти секвенційних *RID*-числень використовується метод модельних множин.

Висновки

У роботі будуються нові логіки, що дозволяють більш адекватно описувати властивості предикатів та функцій, визначених над ієрархічними даними. Характерною властивістю таких логік є використання в їх мовах складних імен. Дається визначення ієрархічних номінативних даних, вводяться та досліджуються операції над такими даними. Будуються композиційно-номінативні логіки часткових предикатів реномінативного та кванторного рівнів над ієрархічними номінативними даними. Вивчаються семантичні властивості цих логік, для таких логік еквітонних квазіарних предикатів реномінативного рівня пропонуються числення секвенційного типу. Дослідження логік квазіарних предикатів над ієрархічними даними кванторного рівня та побудову відповідних секвенційних числень планується продовжити в наступних роботах.

1. *Handbook of Logic in Computer Science*. Edited by S. Abramsky, Dov M. Gabbay and T. S. E. Maibaum.– Oxford University Press.– Vol. 1–5, 1993–2000.
2. Нікітченко Н.С. Композиционно-номинативный подход к уточнению понятия программы // Проблемы програмування.– 1999.– № 1. – С. 16–31.
3. Нікітченко Н.С. Предикатные композиционно-номинативные системы // Проблемы програмування. – 1999. – № 2. – С. 3–19.
4. Нікітченко Н.С., Шкільняк С.С. Неоклассические логики предикатов // Проблемы програмування. – 2000. – № 3–4.
5. Нікітченко М.С., Шкільняк С.С. Математична логіка та теорія алгоритмів. – К., 2008. – 528 с.
6. Басараб І.А., Нікітченко Н.С., Редько В.Н. Композиционные базы данных.– К.: Либідь, 1992.– 191 с.
7. Нікітченко М.С., Шкільняк С.С. Семантичні властивості композиційно-номінативних логік. // Тези доп. Міжнар. конф. "Теоретичні та прикладні аспекти побудови програмних систем" – ТАAPSD'2009.– Київ, 2009. – С. 50–59.