

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

ВІСНИК

**КИЇВСЬКОГО НАЦІОНАЛЬНОГО
УНІВЕРСИТЕТУ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

СЕРІЯ ФІЗИКО-МАТЕМАТИЧНІ НАУКИ

СПЕЦВИПУСК, 2013

**Вісник Київського національного університету імені Тараса Шевченка,
спецвипуск, 2013
Серія фізико-математичні науки**

З 1991 року серії вісників Київського університету “Математика і механіка”, “Фізика”, “Моделирование и оптимизация сложных систем” реорганізовано у “Вісник Київського університету. Серія: фізико-математичні науки”. У віснику містяться результати нових досліджень у різних галузях математики, інформатики, механіки, фізики та радіофізики для наукових працівників, викладачів, аспірантів, інженерів і студентів. Друкується за рекомендаціями Вчених Рад фізичного, радіофізичного, механіко-математичного факультетів та факультету кібернетики.

Журнал “Вісник Київського національного університету імені Тараса Шевченка. Серія фізико-математичні науки” включено до переліку фахових видань ВАК України та реферується в Реферативному журналі та базах даних ВИНІТИ, Росія, Москва.

Редакційна колегія:

Анісімов Анатолій Васильович, чл.-кор. НАН України, д.ф.-м.н., проф., **головний редактор**;
Хусаїнов Денис Яхьєвич, д.ф.-м.н., проф.,
заступник головного редактора, відповідальний за видання;
Акіменко Віталій Володимирович, д.т.н., проф.;
Анісімов Ігор Олексійович, д.ф.-м.н., проф.;
Буй Дмитро Борисович, д.ф.-м.н., проф.;
Булавін Леонід Анатолійович, акад. НАН України, д.ф.-м.н., проф.;
Волошин Олексій Федорович, д.т.н., проф.;
Гарашенко Федір Георгійович, д.т.н., проф.;
Данилов Вадим Васильович, д.ф.-м.н., проф.;
Єжов Станіслав Миколайович, д.ф.-м.н., проф.;
Заславський Володимир Анатолійович, д.т.н., доц.;
Кириченко Володимир Васильович, д.ф.-м.н., проф.;
Козаченко Юрій Васильович, д.ф.-м.н., проф.;
Кудін Володимир Іванович, д.т.н., с.н.с.;
Левитський Сергій Михайлович, д.ф.-м.н., проф.;
Макара Володимир Арсенійович, чл.-кор. НАН України, д.ф.-м.н., проф.;
Макарець Микола Володимирович, д.ф.-м.н., проф.;
Маципура Володимир Тимофійович, д.ф.-м.н., проф.;
Перестюк Микола Олексійович, акад. НАН України, д.ф.-м.н., проф.;
Погорілий Сергій Дем'янович, д.т.н., проф.;
Скришевський Валерій Антонович, д.ф.-м.н., проф.

Редакційний відділ:

Анісімова Тетяна Харитонівна, **відповідальний секретар**;
Безущак Оксана Омелянівна, bezusch@univ.kiev.ua;
Стукаленко Вікторія Віталіївна, stu@univ.kiev.ua;
Родіонова Тетяна Василівна, rodTV@univ.kiev.ua;
Хмелюк Надія Кузьмівна, khmeluk@univ.kiev.ua;
Сільвейструк Людмила Миколаївна, **технічний редактор**, slm-klm@ukr.net.

Адреса редакційної колегії:

Факультет кібернетики, Київський національний університет імені Тараса Шевченка,
пр. Глушкова, 4 д, 03680 Тел. (044) 259-01-49

ISBN 978-966-2142
ISSN 1812-5409

Bakuridze M. Some approximate properties of Cesaro means Fourier series and their conjugates	5
Boyko S.B., Sandrakov G.V. Parameter computing of hydrodynamics processes with phase transitions	11
Didmanidze I., Tkhilaishvili R. Excessive structures to increase the security of the informational systems	17
Ivanov Ie. V. On representation of nondeterministic nonanticipative systems with partially defined inputs and outputs	21
Kolianova T.V. Influence of delay on character of stability of stationary points for mathematical model of purchased immunity	31
Skobelev V.G. Quantum automata with operators that commutes	34
Авраменко О.В., Гуртовий Ю.В., Нарadowий В.В. Оцінка енергії хвильового руху у двошаровій рідині з твердим дном та вільною поверхнею	42
Буй Д.Б., Кахута Н.Д., Сільвейструк Л.М. Загальна теорія бінарних відношень: проекція та відношення сумісності	50
Буй Д.Б., Поляков С.А., Гришко Ю.А. Формальне визначення моделі даних, яка використовується в NoSQL базах даних	56
Власюк А.П., Багнюк О.М. Ідентифікація невідомих параметрів джерел забруднень в двовимірній стаціонарній задачі масопереносу для області з криволінійною межею	60
Глушко І.М. Розширена мультимножинна таблична алгебра: операції з'єднання	68
Гончар С.А. Групова схема Діффі-Хеллмана, яка базується на кучі	72
Дерев'янченко О.В. Сучасні підходи до автоматизованого тестування користувацького інтерфейсу під ОС Android	78
Дорошенко А.Ю., Єгоров В.І., Суслова Л.М. Фреймворк для автоматизованого створення веб-додатків	82
Кривий С.Л., Максимець О.М. Аналіз властивостей мереж Петрі	86
Кудін В.І. Про модель матричної гри у змішаних стратегіях та її аналіз методом допустимих базисних матриць	90
Лавріщева К.М. Підхід до формального подання онтології життєвого циклу програмних систем	94
Лавріщева К.М., Колесник А.Л., Стеняшин А.Ю. Об'єктне-компонентне проектування програмних систем. Теоретичні і прикладні питання	103
Логінов В. PELK: евристичний алгоритм аналізу таблиць ознак	118
Лук'янова О.О. Про бісимуляційну еквівалентність детальної моделі Петрі та її CN-моделі досліджуваної паралельної розподіленої системи	122
Нікітченко М.С., Шкільняк С.С. Композиційно-номінативні логіки із спеціальними предикатами наявності значення для змінних	128
Олецький О.В. Про оптимізацію структури веб-порталу на основі марковських процесів прийняття рішень	134
Омельчук Л.Л. Порівняльний аналіз українського стандарту освітньо-професійної підготовки з інформатики та міжнародних освітніх стандартів	138
Порхун О.В. Огляд базових моделей для вирішення задач кореференс	150
Рубан Н.Н. Использование графовых баз данных при обработке реляционных баз данных	154
Сенченко О.С. Законы де Моргана у табличных алгебрах	158
Слабоспицька О.О. Розвиток моделі властивостей сімейства програмних продуктів для забезпечення їх адаптовності	164
Сугоняк І.І. СППР діагностики захворювань з використанням систем нечіткого вивіду	172
Тарануха В.Ю. Застосування класів, основаних на канонічних формах слів та на граматичних класах, в задачі редукції n-програмної моделі мови для розпізнавання української мови	176
Терещенко В.М. Алгоритми морфінгу полігонів у 2D	180
Тодоріко Б.Д. Схема аналізу властивостей матричної гри в змішаних стратегіях на основі методу псевдобазисних матриць	186
Фісуненко А.Л. Генерація простих багатокутників вирізанням та нарощуванням опуклих оболонки	190
Четирбок П.В. Распознавание объектов с использованием критерия на основе векторной меры близости образов в пространстве ошибок	194
Шкільняк О.С. Секвенційні числення логік часткових предикатів з розширеними реномінаціями	199

**Х міжнародна науково-практична конференція
ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ
ПОБУДОВИ ПРОГРАМНИХ СИСТЕМ
(ТАAPSD'2013)
стислий звіт про проведення**

Конференція проводилася на базі Інституту економіки та управління РВНЗ «Кримський гуманітарний університет» (<http://ieu.kgu.edu.ua/>). Співорганізаторами виступили Міністерство освіти і науки України, Міністерство освіти і науки Автономної Республіки Крим, Республіканський вищий навчальний заклад «Кримський гуманітарний університет», Національний університет «Києво-Могилянська Академія», Кіровоградський державний педагогічний університет імені Володимира Винниченка, Вільнюський державний університет (Литва).

Серед наукових напрямків конференції слід виділити:

- Теорія програмування.
- Формальні методи розробки програмних систем.
- Верифікація і тестування програмних систем.
- Представлення даних.
- Логічні методи обробки комп'ютерних знань.
- Інтелектуальні інформаційні системи.
- Захист та кодування інформації.
- Системні методології.
- Технології та сучасний інструментарій програмування.
- Індустріальні основи виробництва програм.
- Web-програмування та електронна комерція.
- Розподілені інформаційно-обчислювальні системи і середовища.
- Інноваційні методи і технології навчання.
- Бази даних/знань.
- Моделі предметних областей та математичні методи їх дослідження.

Конференція проводилась 25 травня – 2 червня 2013 року.

Кількість учасників:

всього – **47**;

співробітників університету – **16**;

співробітників НАН України – **4**;

провідних учених (академіків, член-кореспондентів, професорів) – **9**;

в т.ч. іногородніх – **14** (*Кримський гуманітарний університет, Ніжинський державний університет, Кіровоградський державний університет, Донецький державний університет; Інститут прикладної математики та механіки (м. Донецьк)*), іноземних – **4** (*Батумський державний університет імені Шота Руставелі, Тбіліський державний університет*).

Програмний комітет конференції – **29** осіб (представники наукових установ України, Грузії, Литви, Казахстану, Німеччини, Словачії). Організаційний комітет – **9** осіб.

Рішення конференції У підведенні підсумків і обговоренні доповідей взяли участь професори, доктори наук Буй Д.Б., Лавріщева К.М., Єршов С.В., Рухая Х.М., Скобелев В.Г. Було відзначено достойний рівень організації конференції, належну забезпеченість технічними засобами, важливість і актуальність тематики доповідей. Серед зауважень та побажань: більше уваги приділяти наступним напрямкам:

- Теорія програмування.
- Формальні методи розробки програмних систем.
- Верифікація і тестування програмних систем.
- Логічні методи обробки комп'ютерних знань.
- Інтелектуальні інформаційні системи.
- Захист та кодування інформації.
- Технології та сучасний інструментарій програмування.
- Індустріальні основи виробництва програм.
- Web-програмування.
- Інноваційні методи і технології навчання.
- Бази даних/знань, бази даних NoSQL.
- Моделі предметних областей та математичні методи їх дослідження.

На основі аналізу, обговорення та дискусії учасників було запропоновано наступне. Провести у 2013 році XI міжнародну наукову конференцію «Теоретичні та прикладні аспекти побудови програмних систем», для чого залучити до участі в засіданнях провідних вітчизняних та закордонних вчених. Запропонувати запрошеним відомим науковцям пленарні розгорнуті доповіді про найбільш актуальні проблеми розвитку інформатики.

Кращі доповіді рекомендувати до опублікування у «Віснику Київського національного університету. Серія фізико-математичні науки».

УДК 517.1

Бакурідзе Мзевинар¹, проф.

Деякі апроксимаційні властивості середніх Чезаро тригонометричних рядів Фур'є та їх сполучних

У труді встановлено деякі нові апроксимаційні
властивості $\sigma_n^a(x, f)$ та $t_n^a(x, f)$.

Ключові слова: Чезаровські середні,
тригонометричний ряд Фур'є, сполучний ряд
Фур'є.

¹ Батумський університет ім. Шота Руставелі,
6010, Грузія, м. Батумі, вул. Ніношвілі, 35.

Mzevinar Bakuridze¹, prof.

Some approximate properties of Cesaro means Fourier series and their conjugates

It is established $\sigma_n^a(x, f)$ and $t_n^a(x, f)$ Cesaro means
some approximate features.

Key words: Chesaro means, Fourier trigonometric
series, conjugated trigonometric series.

¹ Batuni Shota Rustaveli State University, 6010,
Georgia, Batumi, Ninoshvili str., 35

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

Suppose, that $T = [-\pi, \pi]$ and $f : R \rightarrow R$ 2π -
are periodic functions. If $f \in L(T)$ as a rule, $\sigma(f)$
and $\bar{\sigma}(f)$ represent respectively trigonometric
Furrier series and their conjugates

$$\sigma[f](x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos kx + b_k \sin kx, \quad (1)$$

$$\bar{\sigma}[f](x) = \sum_{k=1}^{\infty} -b_k \cos kx + a_k \sin kx, \quad (2)$$

where

$$\left. \begin{aligned} a_k &\equiv a_k(f) = \frac{1}{\pi} \int_T f(t) \cos kt \, dt, \quad k \in N_0 \\ b_k &= b_k(f) = \frac{1}{\pi} \int_T f(t) \sin kt \, dt, \quad k \in N. \end{aligned} \right\} \quad (3)$$

As it is known, the issue of convergence and
summability of series $\bar{\sigma}[f](x)$ are closely related
with corresponding properties of conjugated function
 $\bar{f}$, which is defined with the following equation:

$$\begin{aligned} \bar{f}(x) &= -\frac{1}{2\pi} \lim_{x \rightarrow 0} \int_0^\pi [f(x+t) - f(x-t)] \operatorname{ctg} \frac{t}{2} dt = \\ &= -\frac{1}{2\pi} \int_0^\pi [f(x+t) - f(x-t)] \operatorname{ctg} \frac{t}{2} dt. \end{aligned} \quad (4)$$

As I. I. Privalov showed [1], in $f \in L(T)$

function $\bar{f}$ exists almost everywhere. Let us assign
© M. Bakuridze, 2013

$$\bar{f}_n(x) = -\frac{1}{2\pi} \int_{-\frac{\pi}{n}}^{\frac{\pi}{n}} [f(x+t) - f(x-t)] \operatorname{ctg} \frac{t}{2} dt, \quad n > 1. \quad (5)$$

Assume that $p \in [1, +\infty]$ - is a number. For each
function $f \in L^p(T)$ the following will be considered

$$\|f\|_p = \left\{ \frac{1}{2\pi} \int_T |f(t)|^p dt \right\}^{\frac{1}{p}},$$

And also it will consider the following:

$L^\infty(T) = C(T)$, $\|f\|_c = \|f\|_\infty = \sup_{x \in T} |f(x)|$. Let us set

$$\omega^{(k)}(\sigma, f) = \sup_{|h| \leq \sigma} \left\| \sum_{j=0}^k (-1)^{k+j} \binom{k}{j} f(t + jh) \right\|_p; \quad \sigma \in]0, 2\pi]$$

$\omega^{(k)}(\sigma, f)$ is called module L^p of smoothness of an
order to function f .

In the future we will assume that

$$\omega^{(1)}(\sigma, f)_p \equiv \omega(\sigma, f)_p.$$

Bellow A,

$A(f)$, $A(f, p)$, $A(f, a, p)$, $A(a)$, $A_1(a)$, ... indicate
absolute positive or positive constants depending
only on the specified parameters.

Let ω -be module of continuity. Assume that

$H_p^\omega \equiv H_p^\omega(T) = \{f : \omega(\sigma, f)_p \leq A(f, p)\omega(\sigma)\}$ and

$H_p^\omega = H^\omega$. If $\omega(\sigma) = \sigma^\alpha$, $\alpha \in]0, 1]$, then

$H_p^\omega \equiv Lip(a, p)$. $H^\omega \equiv Lipa$.

With $S_n(x, f)$ and $\bar{S}_n(x, f)$ we will define, respectively partial sums of the series

$$S_n(x, f) = \frac{a_0}{2} + \sum_{k=1}^n a_k \cos kx + b_k \sin ks = \frac{1}{\pi} \int_T f(x+t) D_n(t) dt,$$

(6)

$$\bar{S}_n(x, f) = \sum_{k=1}^n -b_k \cos kx +$$

$$a_k \sin kx = -\frac{1}{\pi} \int_T f(x+t) \bar{D}_n(t) dt,$$

(7)

where $D_n(t)$ - is Dirichlet kernel, and $\bar{D}_n(t)$ - conjugate kernel, i.e.

$$D_n(t) = \frac{1}{2} + \sum_{k=1}^n \cos kt = \frac{\sin\left(n + \frac{1}{2}\right)t}{2 \sin \frac{t}{2}},$$

(8)

$$\bar{D}_n(t) = \sum_{k=1}^n \sin kt = \frac{\cos \frac{t}{2} - \cos\left(n + \frac{1}{2}\right)t}{2 \sin \frac{t}{2}} \quad t \in]0, \pi],$$

(9)

and $D_n(0) = n + \frac{1}{2}$, $\bar{D}_n(0) = 0$, $n \in \mathbb{N}$.

Consider that

$$A_k^a = 1, A_k^a = \frac{(a+1)(a+2)(a+3) \dots (a+k)}{k!},$$

(10)

$k \in \mathbb{N}$, $a > -1$.

It is known (see e.g. A. Sigmund [7], pp. 130-131)

$$\text{that } A_k^a = \sum_{i=0}^k A_{k-i}^{a-1}, \quad A_k^a - A_{k-1}^a = A_k^{a-1}$$

(11),

$$A(a) \leq \frac{A_k^a}{k^a} \leq A_1(a)$$

(12),

$$\text{If } K_n^a(t) = \frac{1}{A_n^a} \sum_{k=0}^n A_{n-k}^{a-1} D_k(t),$$

(13),

$$\tau_n^a(t) = \frac{1}{A_n^a} \sum_{k=0}^n A_{n-k}^{a-1} \bar{D}_k(t),$$

(14)

than they are respectively called Chesaro kernel and its conjugate kernel. It is known that (see e.g. A. Sigmund [2] pp. 157-164), that

$$K_n^a(t) = \varphi_n^a(t) + r_n^a(t),$$

(15)

$$r_n^a(t) = \frac{1}{2} ctg \frac{t}{2} + \psi_n^a(t) + \gamma_n^a(t), \quad t \in]0, \pi],$$

(16)

$$\text{where } \varphi_n^a(t) = \frac{\sin\left[\left(n + \frac{1}{2} + \frac{a}{2}\right)t - \frac{a\pi}{2}\right]}{A_n^a \left(2 \sin \frac{t}{2}\right)^{1+a}},$$

(17)

$$\psi_n^a(t) = \frac{\cos\left[\left(n + \frac{1}{2} + \frac{a}{2}\right)t - \frac{a\pi}{2}\right]}{A_n^a \left(2 \sin \frac{t}{2}\right)^{1+a}},$$

(18)

$$\text{and } \|K_n^a\|_c \leq A(a)n, \quad (19), \quad \|\tau_n^a\|_c \leq A(a)n,$$

(20)

$$|\tau_n^a(t)| \leq \frac{A(a)}{nt^2}, \quad \frac{\pi}{n} \leq t \leq \pi,$$

(21)

$$|\gamma_n^a(t)| \leq \frac{A(a)}{nt^2}, \quad \frac{\pi}{n} \leq t \leq \pi.$$

(22)

In what follows we shall use the Holder inequality for integrals. If $f_1 \in L^p(T)$, $f_2 \in L^q(T)$

and $\frac{1}{p} + \frac{1}{q} = 1$, than

$$\|f_1 f_2\| \leq \|f_1\|_p \|f_2\|_q.$$

(23)

We also use the Minkowski inequality. If

$f_1 \in L^p(T)$, $f_2 \in L^p(T)$, $p \in [1, +\infty[$, than

$$\|f_1 + f_2\|_p \leq \|f_1\|_p + \|f_2\|_p$$

(24)

(see, e.g., Hardy, Littlewood, Polia, p. 179 [3]) the generalized Minkowski inequality is true (under appropriate conditions). If $p \in [1, +\infty[$ is a number, than

$$\left\{ \int_T \left[\int_T |f(x_1, x_2) dx_1|^p dx_2 \right]^{\frac{1}{p}} \leq \int_T \left[\int_T |f(x_1, x_2) dx_1|^p dx_2 \right]^{\frac{1}{p}} dx_1. \right.$$

(25)

Assume that function $f \in L(T)$ than $\sigma_n^a(x, f)$ and $\tau_n^a(x, f)$ symbols denotes the Chesaro means of order $a > -1$ consequently $\sigma[f]$ and $\bar{\sigma}[f]$ i.e.

$$\sigma_n^a(x, f) = \frac{1}{A_n^a} \sum_{k=0}^n A_{n-1}^{a-1} S_k(x, f),$$

$$t_n^a(x, f) = \frac{1}{A_n^a} \sum_{k=0}^n A_{n-1}^{a-1} \bar{S}_k(x, f)$$

where $S_k(x, f)$ and $\bar{S}_k(x, f)$ are given to relations (6) and (7). Using the equalities (13) and (14), we can write

$$\sigma_n^a(x, f) = \frac{1}{\pi} \int_T f(x+t) K_n^a(t) dt, \quad (26)$$

$$t_n^a(x, f) = \frac{1}{\pi} \int_T f(x+t) \bar{K}_n^a(t) dt. \quad (27)$$

Different issues related to the actions $\sigma_n^a(x, f)$ and $t_n^a(x, f)$ averages, are studies in the monograph L.V.Zhizhiashvili. ([4], volumes II-IV).

In this paper we establish some new properties $\sigma_n^a(x, f)$ и $t_n^a(x, f)$. Further assume that

$$g(n, f) = \frac{1}{n} \int_{\frac{\pi}{n}}^{\pi} \frac{\omega(t, f)}{t^2} dt; \quad (28)$$

Here we note that

$$\omega\left(\frac{1}{n}, f\right) \leq g(n, f). \quad (29)$$

Theorem 1. a) Let $a \in (0, 1)$, $p \in (1, +\infty)$ and $ap > 1$. Than for the function $f \in C(T)$ the following inequality is true

$$\|\sigma_n^{-a}(f) - f\|_c \leq A(p, a) n^a \omega\left(\frac{1}{n}, f\right)_p + A(a) g(n, f)$$

b) Let $p \in (1, +\infty)$. Than for the function $f \in C(T)$ the following inequality is true

$$\left\| \sigma_n^{-\frac{1}{p}}(f) - f \right\|_c \leq A(p) \left[n^{\frac{1}{p}} (\ln n)^{1-\frac{1}{p}} \omega\left(\frac{1}{n}, f\right)_p + g(n, f) \right]$$

c) Let $p \in (1, +\infty)$, $a \in (0, 1)$ and $ap \in]0, 1[$ than for the function $f \in C(T)$ the following inequality is true

$$\|\sigma_n^{-a}(f) - f\|_c \leq A(p, a) n^{\frac{1}{p}} \omega\left(\frac{1}{n}, f\right)_p + A(a) g(n, f);$$

Proof.

We will be guided by the method used in the monograph L.V.Zhizhiashvili [4],

Taking into account the relations (8), (11) and (13), conclude that $\frac{2}{\pi} \int_0^{\pi} K_n^{-a}(t) dt = 1$. Therefore,

according to equality (26) we find

$$\sigma_n^{-a}(x, f) - f(x) = \frac{1}{\pi} \int_0^{\pi} \varphi(x, t) K_n^{-a}(t) dt, \quad (30)$$

where

$$\varphi(x, f) = f(x+t) + f(x-t) - 2f(x). \quad (31)$$

By force of (15) and (30) we will have

$$\begin{aligned} \sigma_n^{-a}(x, f) - f(x) &= \\ &= \frac{1}{\pi} \int_0^{\frac{\pi}{n}} \varphi(x, t) K_n^{-a}(t) dt + \frac{1}{\pi} \int_{\frac{\pi}{n}}^{\pi} \varphi(x, t) \varphi_n^{-a}(t) dt + \frac{1}{\pi} \int_{\frac{\pi}{n}}^{\pi} \varphi(x, t) r_n^{-a}(t) dt \equiv \\ &\equiv U_n^{(1)}(x, f, a) + U_n^{(2)}(x, f, a) + U_n^{(3)}(x, f, a) \end{aligned} \quad (32)$$

From which, on the basis of inequality (19) we conclude

$$\|U_n^{(1)}(., f)\|_c \leq A(a) \omega\left(\frac{1}{n}, f\right)_c \quad (33)$$

Further, from the equality (30) considering the relation (21) and symbol (28), we will have

$$\|U_n^{(3)}(., f)\|_c \leq A(a) g(n, f). \quad (34)$$

As

$$\begin{aligned} \sin\left[\left(n + \frac{1}{2} - \frac{a}{2}\right)t - \frac{a\pi}{2}\right] &= \\ &= \sin\left(n + \frac{1}{2} - \frac{a}{2}\right)t \cos \frac{a\pi}{2} - \cos\left(n + \frac{1}{2} - \frac{a}{2}\right)t \sin \frac{a\pi}{2} \end{aligned}$$

Then by force of the relations (12) and (17), from the equality (32), that the evaluation of the experssion $U_n^{(2)}(x, f, a)$ it is sufficient to estimate

$$U_n^{(4)}(x, f, a) = n^a \int_{\frac{\pi}{n}}^{\pi} \varphi(x, t) \frac{\cos\left(\frac{1}{2} - \frac{a}{2}\right)t}{\left(\sin \frac{t}{2}\right)^{1-a}} \sin ntdt \quad (35)$$

$$U_n^{(5)}(x, f, a) = n^a \int_{\frac{\pi}{n}}^{\pi} \varphi(x, t) \frac{\cos\left(\frac{1}{2} - \frac{a}{2}\right)t}{\left(\sin \frac{t}{2}\right)^{1-a}} \cos ntdt \quad (36)$$

Let's consider expression $U_n^{(4)}(x, f, a)$ and consider that

$$V_a(t) \equiv \frac{\cos\left(\frac{1}{2} - \frac{a}{2}\right)t}{\left(\sin \frac{t}{2}\right)^{1-a}} \quad (37)$$

If in the expression $U_n^{(4)}(x, f, a)$, we replace t with $t + \frac{\pi}{n}$ and summarize the received two values

$U_n^{(4)}(x, f, a)$ we will have

$$\begin{aligned} 2U_n^{(4)}(x, f, a) &= n^a \left\{ \int_{\frac{\pi}{n}}^{\pi} \varphi(x, t) V_a(t) \sin ntdt - \int_0^{\frac{\pi}{n}} \varphi\left(x, t + \frac{\pi}{n}\right) V_a\left(t + \frac{\pi}{n}\right) \sin ntdt \right\} = \\ &= n^a \left\{ \int_{\frac{\pi}{n}}^{\pi} \left[\varphi(x, t) - \varphi\left(x, t + \frac{\pi}{n}\right) \right] V_a(t) \sin ntdt \right. \\ &\quad + \int_{\frac{\pi}{n}}^{\pi} \varphi\left(x, t + \frac{\pi}{n}\right) \left[V_a(t) - V_a\left(t + \frac{\pi}{n}\right) \right] \sin ntdt - \\ &\quad \left. - \int_0^{\frac{\pi}{n}} \varphi\left(x, t + \frac{\pi}{n}\right) V_a\left(t + \frac{\pi}{n}\right) \sin ntdt + \right. \\ &\quad \left. + \int_{\frac{\pi}{n}}^{\pi} \varphi(x, t) V_a(t) \sin ntdt \right\} \equiv \\ &\equiv \sum_{i=6}^9 U_n^{(i)}(x, f, a) \end{aligned}$$

Using Helder's inequality, according to the equality (38), we find

$$\begin{aligned} \|U_n^{(6)}(\cdot, f, a)\|_c &\leq \\ &\leq n^a \left\{ \int_{\frac{\pi}{n}}^{\pi} \left| \varphi(x, t) - \varphi\left(x, t + \frac{\pi}{n}\right) \right|^p dt \right\}^{\frac{1}{p}} \left\{ \int_{\frac{\pi}{n}}^{\pi} |V_a(t)|^{\frac{p}{p-1}} dt \right\}^{\frac{p-1}{p}} \quad (39) \end{aligned}$$

If we take into account expression (37) and the fact that $\sin t \geq \frac{2}{\pi}t$, $t \in \left]0, \frac{\pi}{2}\right]$ (40).

We find the following

$$\begin{aligned} \sin\left(n + \frac{1}{2} - \frac{a}{2}\right)t &= \sin nt \cos\left(\frac{1}{2} - \frac{a}{2}\right)t + \cos nt \sin\left(\frac{1}{2} - \frac{a}{2}\right)t, \\ \cos\left(n + \frac{1}{2} - \frac{a}{2}\right)t &= \cos nt \cos\left(\frac{1}{2} - \frac{a}{2}\right)t - \sin nt \sin\left(\frac{1}{2} - \frac{a}{2}\right)t \end{aligned}$$

$$\begin{aligned} &\left\{ \int_{\frac{\pi}{n}}^{\pi} |V_a(t)|^{\frac{p}{p-1}} dt \right\}^{\frac{p-1}{p}} \leq \\ &\leq A(a) \left\{ \int_{\frac{\pi}{n}}^{\pi} t^{-(1-a)\frac{p}{p-1}} dt \right\}^{\frac{p-1}{p}} \leq, \\ &\leq A(p, a) \left(\pi - n^{\frac{1}{p}-a} \right) \quad (41) \end{aligned}$$

if $ap > 1$

$$\left\{ \int_{\frac{\pi}{n}}^{\pi} |V_a(t)|^{\frac{p}{p-1}} dt \right\}^{\frac{p-1}{p}} \leq A(p) (\ln n)^{\frac{p-1}{p}},$$

if $ap = 1$ (42)

and

$$\left\{ \int_{\frac{\pi}{n}}^{\pi} |V_a(t)|^{\frac{p}{p-1}} dt \right\}^{\frac{p-1}{p}} \leq A(p, a) \left(n^{\frac{1}{p}-a} \right), \text{ if } a < \frac{1}{p} \quad (43)$$

than, using the relations (39), (41)-(43), we find

$$\|U_n^{(6)}(., f, a)\|_c \leq \begin{cases} A(p, a)n^a \omega\left(\frac{1}{n}, f\right)_p, & ap > 1 \\ A(p)n^{\frac{1}{p}}(\ln n)^{1-\frac{1}{p}} \omega\left(\frac{1}{n}, f\right)_p, & ap = 1 \\ A(p, a)n^{\frac{1}{p}} \omega\left(\frac{1}{n}, f\right)_p, & ap < 1 \end{cases}$$

It is known that

$$\left|V_a(t) - V_a\left(t + \frac{\pi}{n}\right)\right| \leq \frac{A(a)}{nt^{2-a}}, \quad \frac{\pi}{n} \leq t \leq \pi \quad (46)$$

$$\left|V_a(t) - 2V_a\left(t + \frac{\pi}{n}\right) + V_a\left(t + \frac{2\pi}{n}\right)\right| \leq \frac{A(a)}{n^2 t^{3-a}}, \quad \frac{\pi}{n} \leq t \leq \pi \quad (47)$$

(44) Considering the inequalities (46) and (47), from (45) we will have

Consider expression $U_n^{(7)}(x, f, a)$ from the equality (38). We have

$$U_n^{(7)}(x, f, a) = n^a \left\{ \int_{\frac{\pi}{2n}}^{\frac{2\pi}{n}} \varphi\left(x, t + \frac{\pi}{n}\right) \left[V_a(t) - V_a\left(t + \frac{\pi}{n}\right) \right] \sin dt + \int_{\frac{\pi}{2n}}^{\frac{\pi}{n}} \varphi\left(x, t + \frac{\pi}{n}\right) \left[V_a(t) - V_a\left(t + \frac{\pi}{n}\right) \right] \sin dt \right\}$$

Again by the transformation of the variable t we have

$$U_n^{(7)}(x, f, a) = n^a \left\{ \int_{\frac{\pi}{2n}}^{\frac{2\pi}{n}} \varphi\left(x, t + \frac{\pi}{n}\right) \left[V_a(t) - V_a\left(t + \frac{\pi}{n}\right) \right] \sin dt + \frac{n^a}{2} \int_{\frac{\pi}{2n}}^{\frac{\pi}{n}} \left[\varphi\left(x, t + \frac{\pi}{n}\right) - \varphi\left(x, t + \frac{2\pi}{n}\right) \right] \left[V_a(t) - V_a\left(t + \frac{\pi}{n}\right) \right] \sin ntdt - \frac{n^a}{2} \int_{\frac{\pi}{2n}}^{\frac{2\pi}{n}} \varphi\left(x, t + \frac{2\pi}{n}\right) \left[V_a\left(t + \frac{\pi}{n}\right) - V_a\left(t + \frac{2\pi}{n}\right) \right] \sin ntdt + \frac{n^a}{2} \int_{\frac{\pi}{2n}}^{\frac{\pi}{n}} \varphi\left(x, t + \frac{2\pi}{n}\right) \left[V_a(t) - V_a\left(t + \frac{\pi}{n}\right) \right] \sin ntdt \right\} \equiv \sum_{i=8}^{12} U_n^{(i)}(x, f, a)$$

(45)

$$\|U_n^{(8)}(., f, a)\|_c \leq A(a) \omega\left(\frac{1}{n}, f\right)_c, \quad (48)$$

$$\|U_n^{(9)}(., f, a)\|_c \leq A(a) \omega\left(\frac{1}{n}, f\right)_c \quad (49)$$

$$\|U_n^{(10)}(., f, a)\|_c \leq A(a) n^a \int_{\frac{\pi}{n}}^{\frac{\pi}{2n}} \frac{\omega(t, f)_c}{n^2 t^{3-a}} dt \leq A(a) n^a \omega\left(\frac{1}{n}, f\right)_c \int_{\frac{\pi}{n}}^{\frac{\pi}{2n}} \frac{nt+1}{n^2 t^{3-a}} dt \leq A(a) \omega\left(\frac{1}{n}, f\right)_c \quad (50)$$

$$\|U_n^{(11)}(., f, a)\|_c \leq A(a) \frac{1}{n^{1-a}} \int_{\frac{\pi}{n}}^{\frac{2\pi}{n}} \omega(t, f)_c dt \leq A(a) \omega\left(\frac{1}{n}, f\right)_c \quad (51)$$

$$\|U_n^{(12)}(., f, a)\|_c \leq A(a) n^a \omega\left(\frac{1}{n}, f\right)_c \int_{\frac{\pi}{2n}}^{\frac{\pi}{n}} \frac{dt}{\left(t + \frac{\pi}{n}\right)^{1-a}} \leq A(a) \omega\left(\frac{1}{n}, f\right)_c \quad (52)$$

Accordingly, estimates (45), (48)-(52) give the right to conclude that,

$$\|U_n^{(7)}(., f, a)\|_c \leq A(a) \omega\left(\frac{1}{n}, f\right)_c. \quad (53)$$

If we take into account the equality (38) and the scheme of estimation of the expression $U_n^{(7)}(., f, a)$.

We find that

$$\|U_n^{(i)}(., f, a)\|_c \leq A(a) \omega\left(\frac{1}{n}, f\right)_c \quad (i=13,14)$$

so, from (38), (44), (53) and (54) we find

$$\|U_n^{(4)}(., f, a)\|_c \leq A(p, a) n^a \omega\left(\frac{1}{n}, f\right)_p, \quad ap > 1$$

$$\|U_n^{(4)}(., f, a)\|_c \leq A(p) n^{\frac{1}{p}} (\ln n)^{1-\frac{1}{p}} \omega\left(\frac{1}{n}, f\right)_p, \quad ap = 1 \quad (56)$$

$$\|U_n^{(4)}(., f, a)\|_c \leq A(p, a) n^{\frac{1}{p}} \omega\left(\frac{1}{n}, f\right)_p, \quad ap < 1 \quad (57)$$

Accordingly, relations (32)-(34) and (55)-(57) prove the truth of the theorem 1.

The following is also true

Theorem 2. a) Let $a \in]0, 1[$ - is a number. If function $f \in C(T)$, $a, ap > 1$ than

$$(54) \quad \|t_n^{-a}(f) - \bar{f}_n\|_c \leq A(p, a) n^a \omega\left(\frac{1}{n}, f\right)_p + A(a) g(n, f);$$

b) Let $p \in]1, \infty[$ - is a number, and function $f \in C(T)$. Than

$$(55) \quad \|t_n^{-\frac{1}{p}}(f) - \bar{f}_n\|_c \leq A(p) \left[n^{\frac{1}{p}} (\ln n)^{1-\frac{1}{p}} \omega\left(\frac{1}{n}, f\right)_p + g(n, f) \right].$$

c) Consider that $p \in]0, +\infty[$ - is a number and function $f \in C(T)$. If $ap \in]0, 1[$ than

$$\|t_n^{-a}(f) - \bar{f}_n\|_c \leq A(p, a) n^{\frac{1}{p}} \omega\left(\frac{1}{n}, f\right)_p + A(a) g(n, f), \quad n \geq 4$$

REFERENCES

1. *I.I. Privalov*. Kosh's Integral. Saratov, 1919, (in Russian);
2. *G.G. Xardi D.E. Littlvud. G. Polia*. Inequalities. N., GIL, 1948 (in Russian).
3. *A. Zigmund*. Trigonometric series. T-I, M. Mir 1965 (in Russian).
4. *L. V. Zhizhiashvili*. Some problems about trigonometric series and its conjugates. Tbilisi, TGU, 1993 (in Russian).

Надійшла до редколегії 26.06.2013

УДК 519.6:533.6

Бойко С.Б.¹, старший викладач,
Сандраков Г.В.², д.ф.-м.н., с.н.с.

Обчислювання параметрів процесів гідродинаміки з фазовими переходами

Розглядається новий метод обчислення параметрів динаміки неоднорідної рідини з урахуванням фазових переходів графіт-алмаз. Метод ґрунтується на дискретизації законів збереження маси, моментів і енергії в інтегральній і диференціальній формах. Комбінація методів частинок у комірках Харлоу та методу крупних частинок Білотерковського використовується при чисельній реалізації цього методу. Наведені приклади обчислення параметрів даних процесів.

Ключові слова: моделювання, процеси гідродинаміки, фазові переходи, закони збереження, обчислювання параметрів, перехід графіт-алмаз.

¹ Таврійський державний агротехнологічний університет, 72316, м. Мелітополь, пр. Богдана Хмельницького 18, e-mail: lanamelit@mail.ru

² Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр. Глушкова 4д, e-mail: sandrako@mail.ru

Статтю представив д.т.н., с.н.с. Кудін В.І.

A new method of direct parameter computing for some processes of heterogeneous fluid dynamics with take of phase transitions like graphite-diamond is presented. It is supposed that the fluids are compressible and inviscid (non-viscous). Heterogeneities of the fluids are considered as small drops or particles of one fluid within other fluid. Total number of the drops may be large enough and the drops may have phase transitions. Thus simulations of the main fluid (or gas) with small transited drops dynamics are discussed. These are dynamics of multiphase flows really. Therefore it is possible to use general multiphase flow models in the case. However, relevant equations are not complete as a rule. For example, there is a problem as to distribute energies between the phases in the model dynamics. Various physical experiments are necessary for solving of the problem in concrete cases. The situation is more difficult whenever phase transitions are possible.

S.B. Boyko, senior teacher,
G.V. Sandrakov, d.f.-m.n., senior researcher

Parameter computing of hydrodynamics processes with phase transitions

A new method of parameter computing for heterogeneous fluid dynamics with take of phase transitions like graphite-diamond is presented. The method is based on a discretization of conservation laws for masses, momentums, and energies in integral and differential forms. The combination of Harlow's particle-in-cell method and Belotserkovskii's large particles method is used for parameter computing by the method simulation. Examples of such process parameter computing are presented.

Keywords: modeling, hydrodynamics processes, phase transitions, conservation laws, parameter computing, graphite-diamond transition.

¹ Tavricheskii State Agritechnological University, 72316 Melitopol, Bogdana Hmel'nitskogo av., 18, e-mail: lanamelit@mail.ru

² Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova av., 4d, e-mail: sandrako@mail.ru

Presented method is based on a discretization of conservation laws for masses, momentums, and energies in integral and differential forms. The discretization is natural and numerical simulations are realized as direct computer experiments for dynamics of main fluid together with transited drops without use multiphase flows approach. The method seems to be much more adequate to the physical and mathematical essence of the dynamics because conservation laws are correct on the discrete level at least.

The presented method is a combination of the Harlow's particle-in-cell method and Belotserkovskii's large particles method (see [1] and [2], for example). Let us recall some background of the methods before to give more details on the combination.

Euler's and Lagrange's approaches are used simultaneously in the particles in cells method for homogeneous fluid (or gas). The method is based on

a discretization of conservation laws for masses, momentums, and energies of the fluid in the following integral forms

$$\begin{aligned} \int_{V(t)} \rho' d\tau &= - \int_{S(t)} (\rho W) \cdot N ds, \\ \frac{d}{dt} \int_{V(t)} \rho W d\tau &= - \int_{S(t)} p N ds, \\ \frac{d}{dt} \int_{V(t)} \rho E d\tau &= - \int_{S(t)} (pW) \cdot N ds, \end{aligned} \quad (1)$$

where $V(t)$ and $S(t)$ are volume and surface of some Lagrange's domain in the fluid, N is an exterior normal to the domain, $p = p(\rho, E)$ and ρ, W, E are unknown density, velocity, and full energy. For example, the case of three dimension space may be discussed and therefore $W = (u, v, w)$.

It is known [1] that the conservation are equivalent to conservation laws for masses, momentums, and energies of the fluid in the following differential forms

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \text{div}(\rho W) &= 0, \\ \frac{\partial \rho W}{\partial t} + \text{div}(\rho W \otimes W) + \nabla p &= 0, \\ \frac{\partial \rho E}{\partial t} + \text{div}(\rho E W) + \text{div}(pW) &= 0, \end{aligned} \quad (2)$$

where $W \otimes W$ is the tensor square of vector function $W = W(t, x, y, z)$ and x, y, z is a point of some domain Ω , which is filled by the heterogeneous fluid under consideration.

The time discretization in the method is natural. Simulations are conducted step by step with a small enough time interval Δt that starting from an initial configuration. The space discretization in the method is more complicated and dynamics are taking into consideration. The fluid region Ω is divided into cells with the small size Δx and the fluid filling every such cell is considered as a collection of a few particles or drops. Every such particle have own mass, volume, energy, and coordinates that are specified at an initial moment. In addition the density, velocity, and full energy $\rho^n, u^n, v^n, w^n, E^n$ are specified for every such cell with number i, j, k at the time moment $t = n\Delta t$.

Corresponding time step of the simulation is split up to three stages so the discrete conservation laws are faithful. For example, the total mass of particles under consideration is saved at every time step of such discretization.

On the first stage of the time step, the intermediate velocities $\bar{u}^n, \bar{v}^n, \bar{w}^n$ and the energy $\bar{E}^n$ of particles from the sell with number i, j, k are calculated by the following formulas

$$\begin{aligned} \bar{u}_{i,j,k}^n &= u_{i,j,k}^n - \frac{p_{i+1,j,k}^n - p_{i-1,j,k}^n}{2\Delta x} \frac{\Delta t}{\rho_{i,j,k}^n}, \\ \bar{v}_{i,j,k}^n &= v_{i,j,k}^n - \frac{p_{i,j+1,k}^n - p_{i,j-1,k}^n}{2\Delta y} \frac{\Delta t}{\rho_{i,j,k}^n}, \\ \bar{w}_{i,j,k}^n &= w_{i,j,k}^n - \frac{p_{i,j,k+1}^n - p_{i,j,k-1}^n}{2\Delta z} \frac{\Delta t}{\rho_{i,j,k}^n}, \\ \bar{E}_{i,j,k}^n &= E_{i,j,k}^n - \\ &- \frac{p_{i+1/2,j,k}^n u_{i+1/2,j,k}^n - p_{i-1/2,j,k}^n u_{i-1/2,j,k}^n}{\Delta x} \frac{\Delta t}{\rho_{i,j,k}^n} - \\ &- \frac{p_{i,j+1/2,k}^n u_{i,j+1/2,k}^n - p_{i,j-1/2,k}^n u_{i,j-1/2,k}^n}{\Delta y} \frac{\Delta t}{\rho_{i,j,k}^n} - \\ &- \frac{p_{i,j,k+1/2}^n u_{i,j,k+1/2}^n - p_{i,j,k-1/2}^n u_{i,j,k-1/2}^n}{\Delta z} \frac{\Delta t}{\rho_{i,j,k}^n}, \end{aligned} \quad (3)$$

where

$$\begin{aligned} p_{i+1/2,j,k}^n &= \frac{p_{i+1,j,k}^n - p_{i,j,k}^n}{2}, \quad p_{i-1/2,j,k}^n = \frac{p_{i,j,k}^n - p_{i-1,j,k}^n}{2}, \\ u_{i+1/2,j,k}^n &= \frac{u_{i+1,j,k}^n - u_{i,j,k}^n}{2}, \quad u_{i-1/2,j,k}^n = \frac{u_{i,j,k}^n - u_{i-1,j,k}^n}{2}, \\ v_{i,j+1/2,k}^n &= \frac{v_{i,j+1,k}^n - v_{i,j,k}^n}{2}, \quad v_{i,j-1/2,k}^n = \frac{v_{i,j,k}^n - v_{i,j-1,k}^n}{2} \end{aligned}$$

and similar formula is used for $p_{i,j+1/2,k}^n, \dots, w_{i,j,k-1/2}^n$.

This is the Euler's stage for approximations of transport free momentums equations in (1) by the pressure forces for every sell. On the second stage, motions of the particles by the velocities are taking into account. This is the Lagrange's stage for an approximation of masses equation that is modeling of mass transports from a sell to surrounding sells. On the third stage, moving of the momentums and energy are calculated. This is the concluding stage for approximations of pressure free momentums and energies equations in (1) that is modeling of the momentums and energy transports by the dynamics from a sell to surrounding sells.

The approximations are rationale from physical and mathematical point of view since conservation laws are correct on the discrete levels during the courses of corresponding numerical simulations. Therefore the particle-in-cells method is effective enough for numerical evaluations of homogeneous fluid (or gas) dynamics by boundary conditions and external forces. Concrete types of fluid are defined here by a form of state equation $p = p(\rho, E)$ or $p = p(\rho, J)$, where $J = E - W^2 / 2$ is a notation for interior energy.

An essential problem is only that total number of particles may be very large. Indeed total number of cells must be large enough for best approximations and the number of particles at every cell must be large enough also. Moreover every such particle must have own mass, volume, energy, and coordinates. Thus there is massive data and the data is recalculated from step to step.

In order to avoid the problem it is possibly to use the large particles method. The time discretization in the method is similar to the discretization in the particles in cells method. The space discretization in the method is following. Fluid region is divided into cells with small size and the fluid filling every such cell is considered as a large particle or drop. Every such particle have own mass, volume and energy that are specified at an initial moment. In addition the density, velocity, and full energy are specified for every such cell at the moment. But the volume of particle is coincided with the volume of cell now. Therefore the mass and energy of particle are defined by the density and full energy. Thus the data is not so massive in the method.

Corresponding time step of the simulation is split up to three stages also and so the discrete conservation laws are faithful. The stages are similar to the stages of the particles in cells method, for example, formulas (3) are used on the first stage. Modifications are need only for modeling of the mass and momentums transports by the dynamics. For example, the mass transports are calculated as moving of corresponding share of large particle mass from the cell to corresponding surrounding sell by the following formulas

$$\begin{aligned} M_{i+1/2,j,k}^n &= \bar{\rho}_{i+1/2,j,k}^n \bar{u}_{i+1/2,j,k}^n \Delta y \Delta z \Delta t, \\ M_{i-1/2,j,k}^n &= \bar{\rho}_{i-1/2,j,k}^n \bar{u}_{i-1/2,j,k}^n \Delta y \Delta z \Delta t, \\ M_{i,j+1/2,k}^n &= \bar{\rho}_{i,j+1/2,k}^n \bar{v}_{i,j+1/2,k}^n \Delta x \Delta z \Delta t, \\ M_{i,j-1/2,k}^n &= \bar{\rho}_{i,j-1/2,k}^n \bar{v}_{i,j-1/2,k}^n \Delta x \Delta z \Delta t, \end{aligned} \quad (4)$$

$$M_{i,j,k+1/2}^n = \bar{\rho}_{i,j,k+1/2}^n \bar{w}_{i,j,k+1/2}^n \Delta x \Delta y \Delta t,$$

$$M_{i,j,k-1/2}^n = \bar{\rho}_{i,j,k-1/2}^n \bar{w}_{i,j,k-1/2}^n \Delta x \Delta y \Delta t,$$

where $\bar{u}_{i+1/2,j,k}^n, \bar{u}_{i-1/2,j,k}^n, \bar{v}_{i,j+1/2,k}^n, \dots, \bar{w}_{i,j,k-1/2}^n$ are calculated as in (3) and we use the equalities

$$\begin{aligned} \bar{\rho}_{i+1/2,j,k}^n &= \begin{cases} \rho_{i,j,k}^n, & \text{если } \bar{u}_{i+1/2,j,k}^n \geq 0, \\ \rho_{i+1,j,k}^n, & \text{если } \bar{u}_{i+1/2,j,k}^n < 0, \end{cases} \\ \bar{\rho}_{i-1/2,j,k}^n &= \begin{cases} \rho_{i-1,j,k}^n, & \text{если } \bar{u}_{i-1/2,j,k}^n \geq 0, \\ \rho_{i,j,k}^n, & \text{если } \bar{u}_{i-1/2,j,k}^n < 0, \end{cases} \\ \bar{\rho}_{i,j+1/2,k}^n &= \begin{cases} \rho_{i,j,k}^n, & \text{если } \bar{v}_{i,j+1/2,k}^n \geq 0, \\ \rho_{i,j+1,k}^n, & \text{если } \bar{v}_{i,j+1/2,k}^n < 0, \end{cases} \\ \bar{\rho}_{i,j-1/2,k}^n &= \begin{cases} \rho_{i,j-1,k}^n, & \text{если } \bar{v}_{i,j-1/2,k}^n \geq 0, \\ \rho_{i,j,k}^n, & \text{если } \bar{v}_{i,j-1/2,k}^n < 0, \end{cases} \\ \bar{\rho}_{i,j,k+1/2}^n &= \begin{cases} \rho_{i,j,k}^n, & \text{если } \bar{w}_{i,j,k+1/2}^n \geq 0, \\ \rho_{i,j,k+1}^n, & \text{если } \bar{w}_{i,j,k+1/2}^n < 0, \end{cases} \\ \bar{\rho}_{i,j,k-1/2}^n &= \begin{cases} \rho_{i,j,k-1}^n, & \text{если } \bar{w}_{i,j,k-1/2}^n \geq 0, \\ \rho_{i,j,k}^n, & \text{если } \bar{w}_{i,j,k-1/2}^n < 0. \end{cases} \end{aligned} \quad (5)$$

On the third stage, moving of the momentums $u^{n+1}, v^{n+1}, w^{n+1}$ and the energy E^{n+1} on the step $n+1$ are calculated by the following formulas

$$\begin{aligned} \rho_{i,j,k}^{n+1} &= \rho_{i,j,k}^n - \\ &\quad - O_{xyz}^{-1} \{ M_{i+1/2,j,k}^n - M_{i-1/2,j,k}^n + M_{i,j+1/2,k}^n - \\ &\quad - M_{i,j-1/2,k}^n + M_{i,j,k+1/2}^n - M_{i,j,k-1/2}^n \}, \\ u_{i,j,k}^{n+1} &= \frac{\rho_{i,j,k}^n}{\rho_{i,j,k}^{n+1}} u_{i,j,k}^n - \\ &\quad - O_{xyz}^{-1} \{ \bar{u}_{i+1/2,j,k}^n M_{i+1/2,j,k}^n - \bar{u}_{i-1/2,j,k}^n M_{i-1/2,j,k}^n + \\ &\quad + \bar{u}_{i,j+1/2,k}^n M_{i,j+1/2,k}^n - \bar{u}_{i,j-1/2,k}^n M_{i,j-1/2,k}^n + \\ &\quad + \bar{u}_{i,j,k+1/2}^n M_{i,j,k+1/2}^n - \bar{u}_{i,j,k-1/2}^n M_{i,j,k-1/2}^n \}, \\ v_{i,j,k}^{n+1} &= \frac{\rho_{i,j,k}^n}{\rho_{i,j,k}^{n+1}} v_{i,j,k}^n - \\ &\quad - O_{xyz}^{-1} \{ \bar{v}_{i+1/2,j,k}^n M_{i+1/2,j,k}^n - \bar{v}_{i-1/2,j,k}^n M_{i-1/2,j,k}^n + \\ &\quad + \bar{v}_{i,j+1/2,k}^n M_{i,j+1/2,k}^n - \bar{v}_{i,j-1/2,k}^n M_{i,j-1/2,k}^n + \\ &\quad + \bar{v}_{i,j,k+1/2}^n M_{i,j,k+1/2}^n - \bar{v}_{i,j,k-1/2}^n M_{i,j,k-1/2}^n \}, \end{aligned} \quad (6)$$

$$w_{i,j,k}^{n+1} = \frac{\rho_{i,j,k}^n}{\rho_{i,j,k}^{n+1}} w_{i,j,k}^n -$$

$$- O_{xyz}^{-1} \{ \bar{w}_{i+1/2,j,k}^n M_{i+1/2,j,k}^n - \bar{w}_{i-1/2,j,k}^n M_{i-1/2,j,k}^n +$$

$$+ \bar{w}_{i,j+1/2,k}^n M_{i,j+1/2,k}^n - \bar{w}_{i,j-1/2,k}^n M_{i,j-1/2,k}^n +$$

$$+ \bar{w}_{i,j,k+1/2}^n M_{i,j,k+1/2}^n - \bar{w}_{i,j,k-1/2}^n M_{i,j,k-1/2}^n \},$$

$$E_{i,j,k}^{n+1} = \frac{\rho_{i,j,k}^n}{\rho_{i,j,k}^{n+1}} E_{i,j,k}^n -$$

$$- O_{xyz}^{-1} \{ \bar{E}_{i+1/2,j,k}^n M_{i+1/2,j,k}^n - \bar{E}_{i-1/2,j,k}^n M_{i-1/2,j,k}^n +$$

$$+ \bar{E}_{i,j+1/2,k}^n M_{i,j+1/2,k}^n - \bar{E}_{i,j-1/2,k}^n M_{i,j-1/2,k}^n +$$

$$+ \bar{E}_{i,j,k+1/2}^n M_{i,j,k+1/2}^n - \bar{E}_{i,j,k-1/2}^n M_{i,j,k-1/2}^n \},$$

where $O_{xyz} = \Delta x \Delta y \Delta z$ is the volume of the cell under consideration and the values $\bar{u}_{i+1/2,j,k}^n, \bar{u}_{i-1/2,j,k}^n, \bar{u}_{i,j+1/2,k}^n, \dots, \bar{E}_{i,j,k-1/2}^n$ are calculated as in formulas (5).

Thus total mass of the fluid under consideration is saved at every time step of such discretization if there are no external mass sources under the simulation of course. For example, the mass sources may be induced by boundary conditions and external forces that leads to corresponding modifications of conservation laws in (1) and on the discrete level also. The momentums and energy transports are modeling in similar manners.

The approximations are rationale also. Therefore the large particles method is effective enough for numerical evaluations of homogeneous fluid (or gas) dynamics also and the recalculated data is not very massive as in particle-in-cell method.

Let us return to presented method. The method is designed to numerical modeling of the main fluid with small transited drops dynamics. The time discretization in the method is as in above methods.

The space discretization in the method is following. Heterogeneous fluid region is divided into cells with small size. The main fluid filling every such cell is considered as a large particle while the transited drops are considered as the collection of a few "small" particles in the cell. Every large particle have own mass, volume and energy that are specified at an initial moment. Every small particle have own mass, volume, energy, and coordinates that are specified at the moment. In addition the density, velocity, and full energy are specified for every such cell. This is a combination of above methods at the initial moment.

Corresponding time step of the simulation is split up to three stages with additional preliminary stage.

On the preliminary stage, energies of large particle and small particles in every cell are distributed between the particles so that a pressure in the cell is uniform. Indeed the large particle induce some pressure by own state equation and the small particles induce some pressure by own state equation and it is natural to distribute energies of the particles so that the first pressure coincides with second pressure. Moreover on the stage, it is possible to observe phase transitions of the small particles by the pressure, for example. The phase transitions are realized if the pressure is more than critical pressure by the corresponding phase diagram. In the case the small particles may change own volume, energy, and state equation. Thus the heterogeneous fluid may have three or more phases.

Thus, we use formulas (3) on the stage to calculate the intermediate velocities $\bar{u}^n, \bar{v}^n, \bar{w}^n$ and the energy $\bar{E}^n$ of particles from the sell with number i, j, k at the time moment $t = n\Delta t$.

The remaining stages are similar to the stages of above methods. For example, the mass transports are calculated as moving of corresponding share of large particle mass and small particles masses from the cell to corresponding surrounding sell by formulas (5), which are used in (6).

Thus total mass of the heterogeneous fluid under consideration is saved at every time step of such discretization. The momentums $u^{n+1}, v^{n+1}, w^{n+1}$ and the energy E^{n+1} on the step $n+1$ are modeling in similar manners by formulas (6). Thus this is a combination of above methods from step to step during the courses of corresponding numerical simulations. The method seems to be reasonable for numerical evaluations of such heterogeneous fluid (or gas) dynamics and the recalculated data is not very massive. On the other hand it is possible to use the particles in cells method with the preliminary stages for modeling of the dynamics. But the recalculated data is very massive in the case.

The presented method is designed to numerical modeling of following physical processes. Let consider graphite drops distributing uniformly in some fluid. More exactly, there is heterogeneous medium with graphite particles and the medium may be considered under high pressure as "fluid" with corresponding state equation. For example, we consider a cylinder of the medium that consist of copper with graphite particles. Let the cylinder be in an outside explosive tube device.

Inducing detonation shock waves in the outside explosive tube device, we can observe dynamics of

such shock waves in computer experiments by the method. For example, the pressure dynamics are presented on figures 1 and 2 for time moments 10 and 14 microseconds.

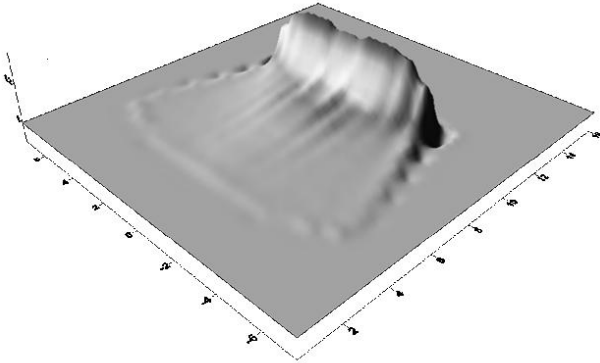


Figure 1. The pressure in outside device
for $t = 10$ microsecond.

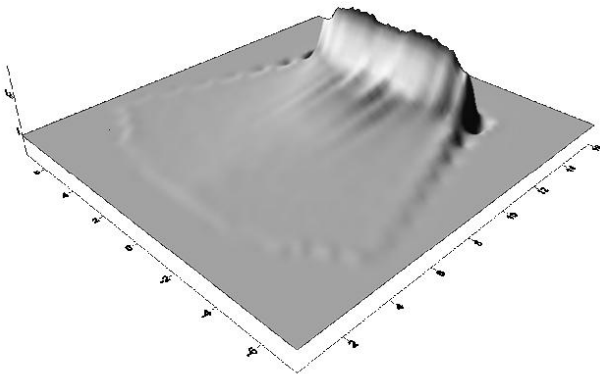


Figure 2. The pressure in outside device
for $t = 14$ microsecond.

The detonation shock waves in the outside device induce conical shock waves in the inside cylinder that is filled by the heterogeneous medium. For example, the pressure dynamics are presented on figures 3,4, and 5 for time moments 10, 14, and 18 microseconds, where the cylindrical coordinates are used. The conical shock waves form concentrating vertex near the cylinder axis.

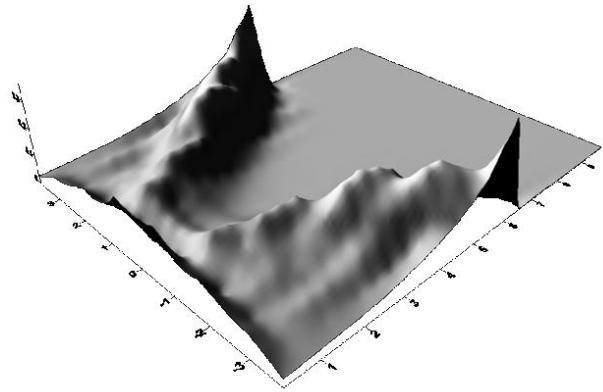


Figure 3. The pressure in inside cylinder
for $t = 10$ microsecond.

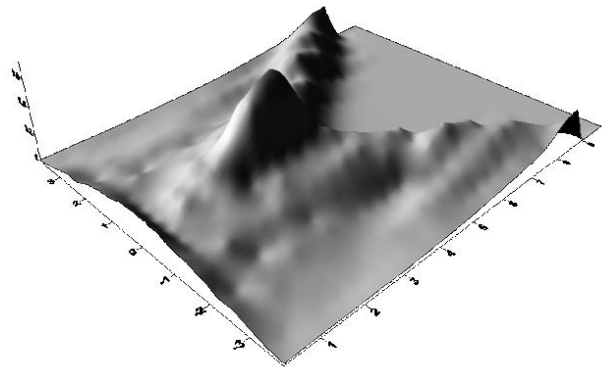


Figure 4. The pressure in inside cylinder
for $t = 14$ microsecond.

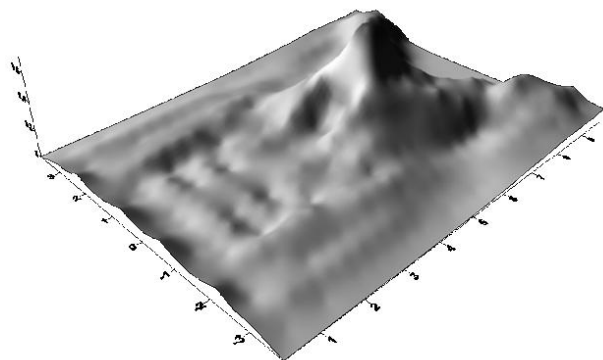


Figure 5. The pressure in inside cylinder
for $t = 18$ microsecond.

The concentrating vertex is named as the Mach disk. The maximal value of the pressure in inside cylinder is near 1.1 GPa for time moments 14 and 18 microseconds on figures 4 and 5. It is known that the critical point of phase transition graphite-diamond is near 0.3 GPa. Thus, it was possible to observe such phase transitions of the graphite particles in computer experiments by the method. Results of the computer experiments were in agreement with results of physical experiments. The results were depending on density of graphite particles and intensity of the conical shock waves. As result, some graphite particle part is transformed to diamond particles. As a rule, the part is near 15%.

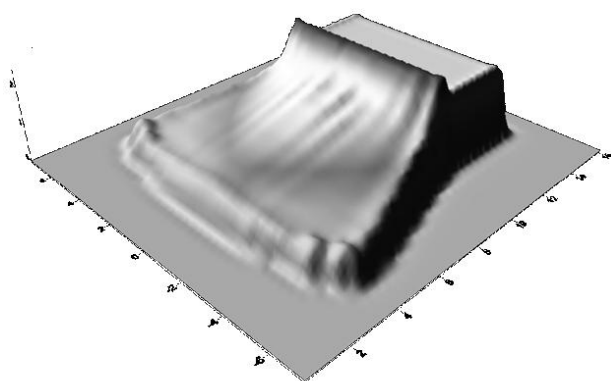


Figure 6. The density in outside device
for $t = 10$ microsecond.

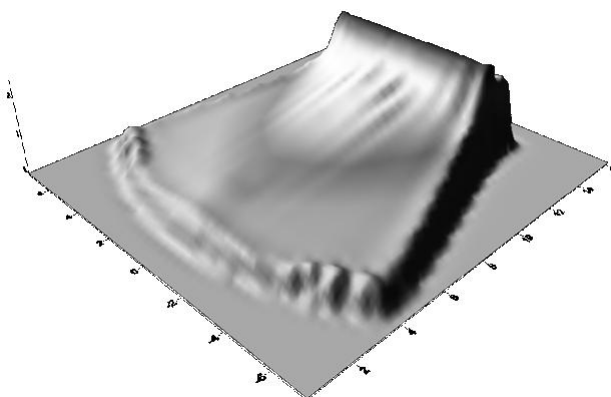


Figure 7. The density in outside device
for $t = 14$ microsecond.

It is known that there arise diamond particles in the outside device also. However, the diamond particles are very small and named as dispersible particles. Really, the particles are diamond nanoparticles and arisen near detonation shock waves in the outside device. For example, the density

dynamics are presented on figures 6 and 7 for time moments 10 and 14 microseconds. The dynamics illustrate detonation shock waves in the outside device. The presented method may be applicable to numerical simulations of the processes.

The presented method was applicable to numerical simulations of plasma dynamics [3]. The plasma may be considered as gas with ionized particles. The gas and particles were defined by corresponding state equations. Equations (2) were coupled with Maxwell's equations and on the discrete level also. Inducing motions of the heterogeneous plasma in some region it was possible to observe absorption of the ionized particles on relevant boundaries in computer experiments by the method coupling with appropriate method for Maxwell's equations [2,3].

The method seems to be perspective for numerical simulations of other absorption and diffusion processes in complex fluid dynamics.

References (List of the used literature)

1. Belotserkovskij O. M. The large particles method in gas dynamics. / O.M. Belotserkovskij, Yu. M. Davidov – Nauka, Moscow, 1982. – 392 p. (in Russian)
2. Shirkov G. Particle-in-cell method for numerical simulation of beam and plasma dynamics / G. Shirkov // Nuclear Instruments and Methods in Physics Research. Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. – 2006. – № 1 (558). – P. 317–324.
3. Boyko S.B. The numerical investigation method for evaporated plasma / S.B. Boyko, V.V. Mischenko, G.V. Sandrakov // Journal Num. and Appl. Math. – 2007. – № 2 (95). – P. 3–12.
4. Sandrakov G.V. Mathematical modeling of complex heterogeneous fluid dynamics / G.V. Sandrakov, S.B. Boyko // Journal Num. and Appl. Math. – 2011. – № 1 (104). – P. 109–120. (in Russian).
5. Boyko S.B. Mathematical modeling of phase transitions graphite-diamond dynamics / S.B. Boyko, G.V. Sandrakov // Journal Num. and Appl. Math. – 2012. – № 2 (108). – P. 88–109. (in Russian).

Надійшла до редколегії 15.07.13

УДК 519.7

Дидманідзе І. Ш.¹, проф.
Тхилаїшвілі Р. Дж.¹, докторант

Надлишкові структури, які збільшують надійність інформаційних систем.

У даній темі обговорюється зв'язок структурної частини інформації з її семантичною стороною, основний сенс комунікаційних систем і за допомогою оптимізації досягнення оптимуму, де були використані кодові структури, було відзначено складність проблеми стиснення та її актуальність.

Такий підхід до питання дає можливість із загальної позиції феномену стиснення зробити поглиблений висновок, як в інформаційному, так і в математичному контексті. Також дає можливість обговорити основні принципи, як в інформаційній, так і в математичній сфері, які аналогічні змістом, але різні формою реалізації.

Ключові слова: Надлишкові структури, надійність інформаційних систем, коригувальні коди.

¹ Батумський університет ім. Шота Руставелі, 6010, Грузія, м. Батумі, вул. Ніношвілі, 35, e-mail: ibraimd@mail.ru

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

Abundance is the warranty of the reliability everywhere. For instance, in the biological structures, in our body (2 eyes, 2 kidneys, the excess of neurons in our brain and etc). In 40-ies of XX century, the famous American scientist, one of the developer of information theory (despite this branch of the science was founded long before) Claude Shannon in his famous theorem showed that unreliable channel (environment) does not limit the reliability of the program transfer, just time for transmission is increased. It means that reliability (stability, steadiness) is ensured by adding the abundance into the transmitted information and this provision spreads not only to the informational systems, but it spreads to the social, biological structures and politics because of its generalization. The last fact was established when it became clear that coding theory is not a mathematical model for the correction of errors, but it is a general model of the organization and management [1].

Didmanidze Ibraim¹, prof.
Tkhilaishvili Rima¹, graduate

Excessive structures to increase the security of the informational systems

This topic discusses the relationship between the structures of the information with the semantic aspect of information, the basic meaning of communication systems and by optimizing the achievement of the optimum, where the structure of the code has been used, it was noted the complexity of the problem of compression and its relevance.

This approach makes it possible to question the position of the general phenomenon of compression to the output as in-depth information and a mathematical context. Also allows the basic principles as discussed in the information and in the mathematical field, which are similar but different content of the form

Keywords: surplus structure, reliability of information systems, error-correcting codes.

¹ Batuni Shota Rustaveli State University, 6010, Georgia, Batumi, Ninoshvili str., 35, e-mail: ibraimd@mail.ru

Let's discuss a specific example. Let's say the binary sequences and symmetrical errors, so the transitions (distortions) are $1\gamma 0$ and $0\gamma 1$, the probability of them is equal. Let us assume that thrice-repeated binary sequence must be transferred into the channel:

$$a_0 = 000, a_1 = 001, a_2 = 010, a_3 = 011, a_4 = 100, \\ a_5 = 101, a_6 = 110, a_7 = 111.$$

There is an unit deviation (error) in the channel which influences with the equal probability to each sequence. They can distort the transferred sequence among three of them. Accordance to above mentioned regulation, it is possible to increase the reliability of system (stability) by adding the proper abundance. Indeed, if we add the initial 3-sequenced message $a_i (i = \overline{0,7})$ to the excess (additional) binary sequences, it is possible to discover (localization)

and correct all the errors. Such structures are called the correcting, extra or coded structures.

The newly constructed (coded) sequences must satisfy the following conditions: the distance between them (any - two sequences (vector) should not be less than the d . In case of the correction of one error - $d \geq 2t \oplus 1$. (addition - mod 2, i.e. what can be divided into 2, is equal to 0). t is the amount of errors. In our case we obtain the significance of coding distance: $d \geq 2t \otimes 13$.

For the next stage, it is necessary that extra coding structure must be built taking into account the d coding distance, which will find and correct all the errors. Clearly we see a binary system [2].

Let's build the proper excessive structure- the corrective code (7.3) for all the errors, where:

3- an initial (informative) sequence;

4- control (excess, extra) sequence;

Proceeding from the above let's build coding operator for code (3x7) matrix [Shannon]

$$L = [E_m B]$$

Where $E_m (m=3)$ is a single matrix (3X3), the part B represents such 4-sequenced binary sequences, where the conditions of coding distance between lines of L matrix is

$$d \geq 3.$$

So the coding matrix must be written as

$$L = \begin{bmatrix} 1000101 \\ 0100011 \\ 0011101 \end{bmatrix}$$

And the decoding matrix will be obtained from

L

$$M = \begin{bmatrix} B \\ E_r \end{bmatrix}, \quad M = \begin{bmatrix} 0101 \\ 0011 \\ 1101 \\ 1000 \\ 0100 \\ 0010 \\ 0001 \end{bmatrix},$$

Where r will be amount of additional (extra) sequences

$$r = n - m = 4.$$

Here n is the total length of code sequence (of code vector, code word), taking into account the additional sequences, m is an initial message, amount of sequences.

Correcting code (7.3) is a space of L - matrix lines or it is a combination of equalities of between these lines including zero sequences: $Q = (0000000)$.

So we have such (7.3) code G :

$$G = \left\{ \begin{array}{l} 100 \ 0101 \\ 010 \ 0011 \\ 001 \ 1101 \\ \hline 110 \ 0110 \\ 101 \ 1000 \\ 011 \ 1110 \\ 111 \ 1101 \end{array} \right\}$$

Essentially V^7 binary, $n = 7$ dimensional vector of space was structured or divided (factorization) into $t=7$ crossing, adjacent classes, where the producing elements for all classes are hindrances $e_i \in E$, where i passes values from 1 to 7 i.e. $e_i \in E (i = \overline{1,7})$.

Hence the building of the excess $r=n-m$ was made from the point of optimum i.e. all the deviations ("from norm") due to discover and localize the hindrances and correct them. As we can see L and M Matrices are interconnected – they are mutually orthogonal, i.e. $LM = 0$, and their construction is very simple. By using this we can make:

- The transfer into 7 dimensioned binary space (vector space V^7) and we have vector V^7 space in the field $GF(2)$ (so vectors of coding elements are 0 or 1);
- The constructed redundant structure allows us to make the organization of the vector space V^7 in a certain manner;
- The Organization (structurisation) allows us to correct all distortion.

So we have obtained the structurisation-organization of - $2^7=128$ sequence in the field $GF(2)$ of V^7 space whose foundation is the sum of 2 (mod2 amount, indicated by $\oplus$) with modul of the lines of L matrix.

So we have the code correcting single errors, where:

e_i - is double indication of single error of designated by 1 in i - sequence;

G - code, (L – length of matrix lines);

M - Matrix of decoding;

t - amount of single errors.

This operation in turn means the division of V^7 space into $t=7$ adjacent classes, each class is represented by the sum mod 2

$$e_i \otimes \{G\} (i = \overline{1,7}).$$

$$e_0 = 000\ 0000 \quad e_1 = 100\ 0000$$

$$\overline{100\ 0101} \quad \overline{000\ 0101}$$

$$010\ 0011 \quad 110\ 0011$$

$$\overline{0011101} \quad \overline{1011101}$$

$$\overline{110\ 0110} \quad \overline{010\ 0110}$$

$$1011000 \quad 0011000$$

$$0111110 \quad 1111110$$

$$1111011 \quad 0111011$$

$$e_2 = 010\ 0000 \quad e_3 = 001\ 0000$$

$$\overline{110\ 0101} \quad \overline{101\ 0101}$$

$$000\ 0011 \quad 001\ 0011$$

$$0111101 \quad 000\ 1101$$

$$\overline{100\ 0110} \quad \overline{111\ 0110}$$

$$1111000 \quad 100\ 1000$$

$$0111110 \quad 010\ 1110$$

$$1011011 \quad 110\ 1011$$

$$e_4 = 000\ 1000 \quad e_5 = 000\ 0100$$

$$\overline{100\ 1101} \quad \overline{100\ 0001}$$

$$010\ 1011 \quad 010\ 0111$$

$$\overline{001\ 0101} \quad \overline{011\ 1001}$$

$$\overline{110\ 1110} \quad \overline{110\ 0010}$$

$$1011000 \quad 1011100$$

$$0110110 \quad 0111010$$

$$1110011 \quad 1111111$$

$$e_6 = 000\ 0010 \quad e_7 = 100\ 0001$$

$$\overline{100\ 0111} \quad \overline{100\ 0100}$$

$$010\ 0001 \quad 010\ 0010$$

$$\overline{0011111} \quad \overline{0011100}$$

$$\overline{110\ 000} \quad \overline{110\ 0111}$$

$$1011010 \quad 1011001$$

$$0111110 \quad 0111111$$

$$1111001 \quad 1111010$$

As mentioned above, the building of L and M matrices provides the the adequate organization of space V7, or the division into eight (7 +1) adjacent classes and the presentation as e_i error or as the producing element of class. $e_0 = (000\ 0000)$ is the leader of zero class or the coded abundance. At the same time, M matrix makes an identification of classes: multiplying of e_i errors by M gives

$e_i M = s_i$ syndrome or the compressed type of i class, indicator which indicates the deviation: $e_i (i = \overline{0,7})$. It provides the localization and identification of e_i error and its correction later [3].

In case of (7,3) correcting code when $n = 7, m = 3, r = 4$ we get the following syndromes as a result of the above mentioned structurisation

$$S_0 = e_0 M = (000\ 0000)M = 000\ 000.$$

$$S_1 = e_1 M = (100\ 0000)M =$$

$$(100\ 0000) \begin{vmatrix} 0101 \\ 0011 \\ 1101 \\ 1000 \\ 0100 \\ 0010 \\ 0001 \end{vmatrix} = 0101.$$

$$S_2 = e_2 M = (010\ 0000)M = (010\ 0000) \begin{vmatrix} B \\ e_r \end{vmatrix} = 0011.$$

$$S_3 = e_3 M = (001\ 0000)M = (001\ 0000) \begin{vmatrix} B \\ e_r \end{vmatrix} = 1101.$$

$$S_4 = e_4 M = (000\ 1000)M = (000\ 1000) \begin{vmatrix} B \\ e_r \end{vmatrix} = 1000.$$

$$S_5 = e_5 M = (010\ 0000)M = (010\ 0000) \begin{vmatrix} B \\ e_r \end{vmatrix} = 0011.$$

$$S_6 = e_6 M = (000\ 0010)M = (000\ 0010) \begin{vmatrix} B \\ e_r \end{vmatrix} = 0010.$$

$$S_7 = e_7 M = (000\ 0001)M = (000\ 0001) \begin{vmatrix} B \\ e_r \end{vmatrix} = 0001.$$

The syndromes of classes and errors are different i.e. the localization of errors must be clearly defined. Classes are not configured and the multiplication of any elements of e_i class by M, so the syndrome for all classes are different and must be $S_i (i = \overline{0,7})$.

Multiplication of element of zero-class by M is equal 0, because it is an element of the matrix core. (There are no errors or mistakes).

For example: $(010\ 0011) \begin{vmatrix} B \\ E_r \end{vmatrix} = 0000 = S_0$.

The multiplication of elements belong to other classes by M is equal to S_i syndrome. Let's take an element of the 3rd class and multiplying by M

$$S_1 = e_1 M = (0111101)M = (0111101) \begin{vmatrix} 0101 \\ 0011 \\ 1101 \\ 1000 \\ 0100 \\ 0010 \\ 0001 \end{vmatrix}$$

= 0011.

Indeed, S_2 syndrome indicates to third class and the error into the second sequence. So the error

BYBLIOGRAPHY

1. *Didmanidze I., Tkilaishvili R.* Data reduction methods. // Proceedings of the II International Scientific and Practical Internet Modern Tendencies of Mathematics and Its Applied – 2013 ~. pp.92-93. Donetsk. 2013.

$e_2 = (010\ 0000)$ is localized and for the correction, it is necessary to implement the change “1” with “0” in the second rank of the obtained sequence or that is the same with module, 2 plus 1.

$$\begin{array}{r} 0111101 \\ \otimes 010\ 0000 \\ \hline 0011101 \end{array}$$

This is the corrected coded word. The localization and correction of the error is obtained in certain forms, by carrying out the organization or adding some excess into the initial message.

2. *Kifshidze Z., Ananiashvili G., Qavtaradze N.* About systematic presentation of adoption of the knowledge, mag. “Intellect”, Tbilisi 2012
3. *Nanobashvili N.* Wholeness of from Perception at Multilevel (Heirarchic) coding and its Appendices Bull. //Of the Academy of Sciences of the Georgian №4, 2003

Надійшла до редколегії 26.06.2013

УДК 004.4

Іванов Є.В.¹, аспірант

Про представлення недетермінованих причинних систем з частково визначеними входами і виходами

У роботі розглянуто клас абстрактних недетермінованих моделей систем типу вхід-вихід з частково визначеними входами і виходами. Моделі даного класу називаються блоками. У роботі показано, що кожен блок, що задовольняє сильній умові причинності має представлення у вигляді абстрактної динамічної системи спеціального вигляду.

Ключові слова: система вхід-вихід, стан, представлення, причинність, недетермінізм.

¹ Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Глушкова 4д, e-mail: ivanov.eugen@gmail.com

Ie. V. Ivanov¹, PhD student

On representation of nondeterministic nonanticipative systems with partially defined inputs and outputs

In the paper a class of abstract nondeterministic input-output models with partially defined inputs and outputs is considered. The models of this class are called blocks. It is shown that each block which is nonanticipative in a strong sense has a representation in the form of an abstract dynamical system of a special kind.

Key Words: input-output system, state, representation, nonanticipation, nondeterminism.

¹ Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova st., 4d, e-mail: ivanov.eugen@gmail.com

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

1. Introduction

Many software tools for developing control and signal processing systems, e.g. *Simulink*, *SCADE*, *Scicos* [1], are based on block diagram notations familiar to control engineers. In these notations a system is described by a diagram consisting of blocks (components) connected by directed links. The blocks describe transformations of input signals (time-varying quantities) into output signals.

In system design tools the topics of automatic analysis and verification are now becoming increasingly important [2]. This makes it necessary to have a rigorous semantic foundation for block diagram notations. Control theory, computer science, and different variants of mathematical systems theory [3-5] all provide some degree of formal treatment of blocks and block diagrams, but their methods are usually not sufficient to handle some important aspects of system design languages such as partially defined block operations, nondeterminism, mixing of analog and discrete-time blocks, non-numeric data processing, etc.

In the previous work [6] we introduced abstract formal models of blocks and block diagrams which overcome these limitations. These models support partially defined inputs and outputs and nondeterminism and are not restricted to a particular

kind of dynamic behavior (e.g. only discrete or only continuous). In particular, we defined a formal notion of a block as an object with an associated multifunction (operation) which maps collections of input signals to collections of output signals. This corresponds to an abstract input-output view of a component of a system (or a whole system).

However, there exists an alternative view of the behavior a system or component which utilizes the concept of state. With this approach a block can be considered as a kind of dynamical system in which state evolves in time according to a certain rule. Many variants of mathematical systems theory [3-5] consider both views of a system (component), i.e. input-output view and a state evolution view and establish a link between them. An essential part of this link is usually a representation theorem which states that a particular input-output model has a representation in the form of a dynamical system of a certain type.

In this work we continue development of the formalism introduced in [6] and establish a representation theorem for our notion of block in the form of an abstract dynamical system of a special kind, namely a Nondeterministic Complete Markovian System (NCMS), which was introduced in the work [7].

2. Notation

We will use the following notation: $\mathbf{R}$ is the set of real numbers, $f: A \rightarrow B$ is a total function from A to B , $f: A \rightharpoonup B$ is a partial function from A to B , 2^A is the power set of a set A , $f|_X$ is the restriction of a function f onto a set X . If A, B are sets, then B^A denotes the set of all total functions from A to B . For a function $f: A \rightharpoonup B$ the symbol $f(x) \downarrow$ ($f(x) \uparrow$) means that $f(x)$ is defined (respectively undefined) on the argument x . The symbols $\neg, \vee, \wedge$ denote the logical negation, disjunction and conjunction respectively. The symbols $\sup$ and $\inf$ denote supremum and infimum. The symbol $\circ$ denotes the functional composition: $(f \circ g)(x) = f(g(x))$. The domain of a function is denoted as $\text{dom}(f) = \{x \mid f(x) \downarrow\}$.

Denote by T the positive real time scale $[0, +\infty)$. We assume that T is equipped with a topology induced by the standard topology on real numbers. Additionally, we define the following set:

$$T_0 = \{\emptyset\} \cup \{[0, t] \mid t \in T \setminus \{0\}\} \cup \{[0, t] \mid t \in T\}.$$

We will need the following notions of a multi-valued function and a named set:

Definition 1 [8]. A (total) multi-valued function from a set A to B (denoted as $f: A \rightarrow^{tm} B$) is a function $f: A \rightarrow 2^B \setminus \{\emptyset\}$.

Definition 2 [8]. A named set is a partial function $f: V \rightharpoonup W$ from a non-empty set of names V to a set of values W .

We will use a special notation for the set of named sets: ${}^V W$ denotes the set of all named sets $f: V \rightharpoonup W$. Named sets are equal, if their graphs are equal. An expression of the form $[n_1 \mapsto a_1, n_2 \mapsto a_2, \dots]$ (where $n_1, n_2, \dots$ are distinct names) denotes a named set d such that the graph of d is $\{(n_1, a_1), (n_2, a_2), \dots\}$. The nowhere-defined named set is called the empty named set and is denoted as $[\]$.

3. An abstract block

In this section we recall some definitions introduced in the paper [6].

Informally, a block (see Fig. 1) is a device which receives input signals and produces output signals. We call a collection of input (output) signals an input (respectively, output) signal bunch.

At each time moment (the value of) a given signal may be present or absent. In the general case, the presence of an input signal at a given time does not imply the presence of an output at the same or any other time moment.

A block may operate non-deterministically, i.e. for one input signal bunch it may choose an output

signal bunch from a set of possible variants. However, for each input signal bunch there exists at least one corresponding output signal bunch (though the values of all signals in it may be absent at all times which means that the block does not produce output values).

Normally, a block processes the whole input signal bunch and does or does not produce output values. But, in the general case, a block may not process the whole input signal bunch and may terminate before its end. This can be interpreted as an abnormal situation (e.g., caused by invalid input).

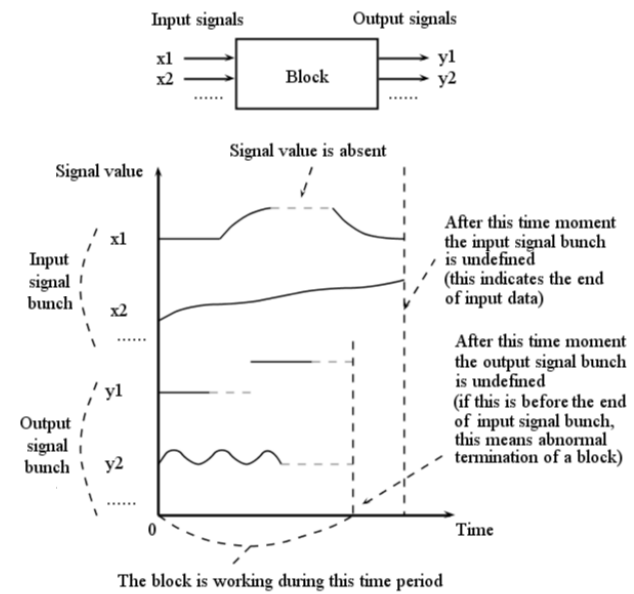


Fig. 1. An illustration of a block. The plot displays example evolutions of input and output signals. The input and output signals are lumped into an input and output signal bunch respectively. Solid curves represent (present) signal values. Dashed horizontal segments indicate signal absence. Dashed vertical lines indicate the right boundaries of the domains of signal bunches

Let W be a non-empty set of values.

Definition 3. [6]

- (1) A (partial) V -signal bunch (where V is a set of names) is a function $s: T \rightharpoonup {}^V W$ such that $\text{dom}(s) \in T_0$. The set of all V -signal bunches is denoted as $Sb(V, W)$.
- (2) A signal bunch is a partial V -signal bunch for some V .
- (3) A signal bunch s is called trivial (denoted as $\perp$), if $\text{dom}(s) = \emptyset$ and is called total, if $\text{dom}(s) = T$.
- (4) For a signal bunch s , a signal corresponding to a name x is a partial function $t \mapsto s(t)(x)$. It is denoted as $s[x]$.

- (5) A signal bunch s_1 is a prefix of a signal bunch s_2 (denoted as $s_1 \sqsubseteq s_2$), if $s_1 = s_2|_A$ for some $A \in T_0$. Note that $\sqsubseteq$ on V -signal bunches is a partial order.

Definition 4. [6]

- (1) A block is an object B together with an associated set of input names $In(B)$, a set of output names $Out(B)$, and a total multi-valued function $Op(B) : Sb(In(B), W) \rightarrow^{tm} Sb(Out(B), W)$ such that $o \in Op(B)(i)$ implies $dom(o) \subseteq dom(i)$.
- (2) Two blocks B_1, B_2 are called semantically identical, if $In(B_1) = In(B_2)$, $Out(B_1) = Out(B_2)$, and $Op(B_1) = Op(B_2)$.
- (3) An I/O pair of a block B is a pair of signal bunches (i, o) such that $o \in Op(B)(i)$. The set of all I/O pairs of B is denoted as $IO(B)$ and is called the input-output (I/O) relation of B .

An inclusion $o \in Op(B)(i)$ means that o is a possible output of a block B on the input i . For each input signal bunch i there is some output signal bunch o . The domain of o is a subset of the domain of i . A situation when o becomes undefined at some time t , but i is still defined at t can be interpreted as a fatal error during block's operation (the block cannot resume its operation after t).

Definition 5. [6] A block B is deterministic, if $Op(B)(i)$ is a singleton set for each i .

An important aspect of blocks which can be implemented as devices working in the real (physical) time is causality (nonanticipation).

Definition 6. [6] A deterministic block B is causal, if for all signal bunches $i_1, i_2, A \in T_0$, and $o_1 \in Op(B)(i_1), o_2 \in Op(B)(i_2)$, the equality $i_1|_A = i_2|_A$ implies $o_1|_A = o_2|_A$.

Definition 7. [6] A block B is a sub-block of a block B' (denoted as $B \trianglelefteq B'$), if $In(B) = In(B')$, $Out(B) = Out(B')$, and $IO(B) \subseteq IO(B')$.

Definition 8. [6] A block B is strongly nonanticipative, if for each pair $(i, o) \in IO(B)$ there exists a causal deterministic sub-block $B' \trianglelefteq B$ such that $(i, o) \in IO(B')$.

Informally, the operation of a strongly nonanticipative block B can be interpreted as a two-step process:

1) before receiving the input signals, the block B (nondeterministically) chooses a deterministic causal sub-block B' (response strategy);

2) the block B' receives input signals of B and produces the corresponding output signals (response) which become the output signals of B .

4. A Nondeterministic Complete Markovian

System (NCMS)

In this section we recall some definitions introduced in the work [7].

Denote by $\mathbf{T}$ the set of all connected subsets of T (i.e. bounded and unbounded intervals) with cardinality greater than one.

Let Q be a set (a state space) and Tr be some set of functions of the form $s : A \rightarrow Q$, where $A \in \mathbf{T}$. Let us call its elements (partial) trajectories.

Definition 9. [7] A set of trajectories Tr is closed under proper restrictions (CPR), if $s|_A \in Tr$ for each $s \in Tr$ and $A \in \mathbf{T}$ such that $A \subseteq dom(s)$.

Definition 10. [7]

- (1) A trajectory $s_1 \in Tr$ is a subtrajectory of $s_2 \in Tr$ (denoted as $s_1 \sqsubseteq s_2$), if $s_1 = s_2|_{dom(s_1)}$.
- (2) A trajectory $s_1 \in Tr$ is a proper subtrajectory of $s_2 \in Tr$ (denoted as $s_1 \sqsubset s_2$), if $s_1 \sqsubseteq s_2$ and $s_1 \neq s_2$.
- (3) Trajectories $s_1, s_2 \in Tr$ are incomparable, if neither $s_1 \sqsubseteq s_2$, nor $s_2 \sqsubseteq s_1$.

Obviously, $(Tr, \sqsubseteq)$ is a (possibly empty) partially ordered set (poset).

Definition 11. [7] A CPR set of trajectories Tr is

- (1) Markovian, if for each $s_1, s_2 \in Tr$ and $t \in T$ such that $t = \sup dom(s_1) = \inf dom(s_2)$, $s_1(t) \downarrow, s_2(t) \downarrow$, and $s_1(t) = s_2(t)$, the following function s belongs to $Tr : s(t) = s_1(t)$, if $t \in dom(s_1)$, and $s(t) = s_2(t)$, if $t \in dom(s_2)$.
- (2) complete, if each non-empty chain in $(Tr, \sqsubseteq)$ has a supremum.

Definition 12. [7] A nondeterministic complete Markovian system (NCMS) is a triple (T, Q, Tr) , where Q is a set (state space) and Tr (trajectories) is a set of functions of the form $s : A \rightarrow Q, A \in \mathbf{T}$, which is CPR, complete, and Markovian.

The notion of NCMS is close to the notion of a solution system in the sense of Theory of Processes by O. Hájek [9, Definition 2.1], but there are some differences, among which the most significant is the Completeness property of NCMS which is not assumed in the (stand-alone, i.e. independent of the notion of a process) definition of a solution system.

5. Main result

In this section we will introduce a representation of strongly nonanticipative blocks using NCMS.

Definition 13. An input-output (I/O) NCMS is an NCMS (T, Q, Tr) such that Q has a form ${}^I W \times X \times {}^O W$ for some sets I (set of input names), $X \neq \emptyset$ (set of internal states), and O (set of output names). The ${}^I W$ is called an input data set and ${}^O W$ is called an output data set.

Informally, an I/O NCMS describes possible evolutions (trajectories) of triples (d_{in}, x, d_{out}) of input data ($d_{in} \in {}^I W$), internal state ($x \in X$), and output data $d_{out} \in {}^O W$.

It follows immediately from the definition that each I/O NCMS (T, Q, Tr) has a unique set of input names, internal states, and output names. We will denote these sets as $In(\Sigma)$, $IState(\Sigma)$, and $Out(\Sigma)$ respectively.

For any I/O NCMS $\Sigma = (T, Q, Tr)$ and a state $q \in Q$ we will denote as $in(q)$, $istate(q)$, $out(q)$ the projections of q on the first, second, and third coordinate respectively. Correspondingly, for any $s \in Tr$, $in \circ s$, $istate \circ s$, $out \circ s$ denote the composition of the respective projection map with a trajectory.

For each $i \in Sb(In(\Sigma), W)$ let us denote

$$S(\Sigma, i) = \{s \in Tr \mid dom(s) \in T_0 \wedge in \circ s \leq i\};$$

$S_{max}(\Sigma, i)$ is the set of all $\sqsubseteq$ -maximal (i.e. non-continuable) trajectories from $S(\Sigma, i)$;

$$S_{init}(\Sigma, i) = \{s(0) \mid s \in S(\Sigma, i)\};$$

$$S_{init}(\Sigma) = \{s(0) \mid s \in Tr \wedge dom(s) \in T_0\}.$$

For each $Q' \subseteq Q$ let us denote:

$Sel_{1,2}(Q', d, x) = \{q \in Q' \mid \exists d' q = (d, x, d')\}$, i.e., selection of states from Q' by the value of the first and second component.

For each $Q' \subseteq Q$ and $i \in Sb(In(\Sigma), W)$ denote:

$$o_{all}(\Sigma, Q', i) = \{\perp\}, \text{ if } Q' = \emptyset \text{ or } i = \perp;$$

$o_{all}(\Sigma, Q', i) = \{\{0\} \mapsto out(q) \mid q \in Q'\}$, if $Q' \neq \emptyset$ and $dom(i) = \{0\}$;

$o_{all}(\Sigma, Q', i) = \{out \circ s \mid s \in S_{max}(\Sigma, i) \wedge s(0) \in Q'\} \cup \{\{0\} \mapsto out(q) \mid q \in Q' \setminus S_{init}(\Sigma, i)\}$, if $Q' \neq \emptyset$ and $\{0\} \subset dom(i)$, where $\{0\} \mapsto out(q)$ denotes a signal bunch defined on $\{0\}$ which takes a value $out(q)$.

Also, for each $Q_0 \subseteq Q$ let us denote:

$$O_{all}(\Sigma, Q_0, i) = \{\perp\}, \text{ if } dom(i) = \emptyset;$$

$$O_{all}(\Sigma, Q_0, i) = \bigcup_{x \in IState(\Sigma)} o_{all}(\Sigma, Sel_{1,2}(Q_0, i(0), x), i),$$

if $dom(i) \neq \emptyset$.

Definition 14. An initial I/O NCMS is a pair (Σ, Q_0) such that $\Sigma = (T, Q, Tr)$ is a I/O NCMS and Q_0 is a set (of initial states) such that $S_{init}(\Sigma) \subseteq Q_0 \subseteq Q$.

Definition 15. A NCMS representation of a block B is an initial I/O NCMS (Σ, Q_0) such that $In(B) = In(\Sigma)$, $Out(B) = Out(\Sigma)$, and $Op(B)(i) = O_{all}(\Sigma, Q_0, i)$ for all $i \in Sb(In(B), W)$.

Informally, the operation of a block B represented by an initial I/O NCMS (Σ, Q_0) on an input signal bunch i can be described as follows:

1) If $i(0)$ is undefined, then B stops (the output signal bunch is $\perp$).

2) Otherwise, B chooses an arbitrary internal state $x \in IState(\Sigma)$.

3) If there is no admissible initial state $q \in Q_0$ with $in(q) = i(0)$ and $istate(q) = x$ (i.e. $Sel_{1,2}(Q_0, i(0), x) = \emptyset$), then B stops.

4) Otherwise, B chooses an arbitrary $q \in Q_0$ such that $in(q) = i(0)$ and $istate(q) = x$ (i.e. $q \in Sel_{1,2}(Q_0, i(0), x)$).

5) If $dom(i) = \{0\}$ or there is no trajectory s which starts in q and is defined on some interval (of positive length) from T_0 , then B outputs $out(q)$ at time 0 and stops.

6) Otherwise, B chooses an arbitrary maximal trajectory s defined on an interval from T_0 such that $s(0) = q$ and $in \circ s \leq i$ and outputs the signal bunch $out \circ s$.

Theorem 1. Each strongly nonanticipative block has a NCMS representation.

We will prove this theorem in the next section.

6. Proof of Theorem 1

Let us prove several helper lemmas.

Definition 16. A state-restriction of a NCMS $\Sigma = (T, Q, Tr)$ on a set U , denoted as $\Sigma|_U$, is a triple $(T, Q \cap U, \{s \in Tr \mid \forall t \in dom(s) s(t) \in U\})$.

Lemma 1. $\Sigma|_U$ is a NCMS.

Proof follows immediately from Definition 12.

Lemma 2. Let $J \neq \emptyset$ and $((T, Q_j, Tr_j))_{j \in J}$ be an indexed family of NCMS. Then (T, Q, Tr) is a NCMS, where $Q = \bigcap_{j \in J} Q_j$ and $Tr = \bigcap_{j \in J} Tr_j$.

Proof follows immediately from Definition 12.

Lemma 3. Let (T, Q, Tr) be a NCMS, Q' be a set, $f: Q \rightarrow Q'$ be an injective function, and $Tr' = \{f \circ s \mid s \in Tr\}$. Then (T, Q', Tr') is a NCMS.

Proof.

Let us show that Tr' is closed under proper restrictions (CPR). Let $s' \in Tr'$, $A \in \mathbf{T}$, and $A \subseteq dom(s)$. Then $s' = f \circ s$ for some $s \in Tr$, whence $s|_A \in Tr$, because Tr is CPR and $dom(s) = dom(s')$. Thus $s'|_A = f \circ (s|_A) \in Tr$.

Let us show that Tr' is Markovian. Let $s_1', s_2' \in Tr'$, $t^* = \max dom(s_1') = \min dom(s_2')$, and $s_1'(t^*) = s_2'(t^*)$. Then $s_1' = f \circ s_1$, $s_2' = f \circ s_2$ for some $s_1, s_2 \in Tr$. Then $f(s_1(t^*)) = f(s_2(t^*))$, whence $s_1(t^*) = s_2(t^*)$, because f is injective. Then a function $s: dom(s_1) \cup dom(s_2) \rightarrow Q$ such that $s(t) = s_1(t)$ if $t \in dom(s_1)$ and $s(t) = s_2(t)$, if $t \in dom(s_2)$ belongs to Tr . Then $s' = f \circ s \in Tr'$ and $s'(t) = s_1'(t)$, if $t \in dom(s_1')$ and $s'(t) = s_2'(t)$, if $t \in dom(s_2')$.

Let us show that Tr' is complete. Let $c' \subseteq Tr'$ be a non-empty $\sqsubseteq$ -chain. Let $c = \{s \in Tr \mid f \circ s \in c'\} \subseteq Tr$. If $s_1, s_2 \in c$, then $f \circ s_1 \sqsubseteq f \circ s_2$, or $f \circ s_2 \sqsubseteq f \circ s_1$, whence $s_1 \sqsubseteq s_2$ or $s_2 \sqsubseteq s_1$, because f is injective. Thus c is a $\sqsubseteq$ -chain. It is non-empty, because $c' \neq \emptyset$ and s such there exists $s \in Tr$ that $f \circ s = s'$ for any $s' \in c'$. Then there exists a least upper bound $s \in Tr$ of c (when Tr is viewed as poset with respect to $\sqsubseteq$). Let $s^* = f \circ s$. Then $s^* \in Tr$, $s' \sqsubseteq s^*$ for all $s' \in c'$ and the graph of s^* is the union of the graphs of the members of c' . Thus s^* is the least upper bound of c' (when Tr' is viewed as poset with respect to $\sqsubseteq$).

Lemma is proved.

Lemma 4. Let (T, Q^j, Tr^j) , $j \in J$ be an indexed family of NCMS such that $Q^j \cap Q^{j'} = \emptyset$, if $j \neq j'$. Let $Q = \bigcup_{j \in J} Q^j$ and $Tr = \bigcup_{j \in J} Tr^j$.

Then (T, Q, Tr) is a NCMS.

Proof.

Let us show that Tr' is closed under proper restrictions (CPR). Let $s \in Tr$, $A \in \mathbf{T}$, and $A \subseteq dom(s)$. Then $s \in Tr^j$ for some $j \in J$, whence $s|_A \in Tr^j \subseteq Tr$.

Let us show that Tr is Markovian. Let $s_1, s_2 \in Tr$, $t^* = \max dom(s_1) = \min dom(s_2)$, and $s_1(t^*) = s_2(t^*)$. Then $s_1 \in Tr^j$ and $s_2 \in Tr^{j'}$ for some $j, j' \in J$. Then $j = j'$, because otherwise, $Q^j \cap Q^{j'} = \emptyset$. Then a function $s : dom(s_1) \cup dom(s_2) \rightarrow Q$ such that $s(t) = s_1(t)$, if $t \in dom(s_1)$ and $s(t) = s_2(t)$, if $t \in dom(s_2)$ belongs to $Tr^j \subseteq Tr$.

Let us show that Tr is complete. This is obvious, if $Tr = \emptyset$, so assume that $Tr \neq \emptyset$. Let $c \subseteq Tr$ be a non-empty $\sqsubseteq$ -chain. For each $s \in Tr$ there exists an index $j(s) \in J$ such that $s \in Tr^{j(s)}$. For each $s_1, s_2 \in c$, either $s_1 \sqsubseteq s_2$, or $s_2 \sqsubseteq s_1$, and because $dom(s_1), dom(s_2) \neq \emptyset$ and the sets Q^j are disjoint for different j , we have $j(s_1) = j(s_2)$. Thus all indices $j(s)$, $s \in c$ coincide, so there exists $j \in J$ such that $c \subseteq Tr^j$. Then there exists a least upper bound $s^* \in Tr^j$ of c in sense of the poset Tr^j (with the ordering $\sqsubseteq$). Then it is easy to see that s^* is a least upper bound of c in sense of the poset Tr (with the ordering $\sqsubseteq$). Lemma is proved.

Lemma 5. Let Σ be a I/O NCMS, $i \in Sb(In(\Sigma), W)$, and $s \in S(\Sigma, i)$. Then there exists $s' \in S_{max}(\Sigma, i)$ such that $s \sqsubseteq s'$.

Proof.

Consider a set $G = \{s'' \in S(\Sigma, i) \mid s \sqsubseteq s''\}$.

Let $c \subseteq G$ is a non-empty $\sqsubseteq$ -chain. Then it has a least upper bound s^* in Tr , because Σ is a NCMS. This implies that $A \subseteq dom(s^*)$ for some $A \in \mathbf{T}_0 \setminus \{\emptyset\}$, whence $0 \in dom(s^*)$. Then $dom(s^*) \in \mathbf{T}_0$, because $s^* \in Tr$. Moreover, $in \circ s^* \leq i$, because $in \circ s'' \leq i$ for all $s'' \in c$. Then $s^* \in S(\Sigma, i)$. Obviously, $s \sqsubseteq s^*$, so $s^* \in G$.

Thus each non-empty $\sqsubseteq$ -chain of elements of G has a least upper bound in the poset $(Tr, \sqsubseteq)$ which belongs to G . Because $s \in G$, we have $G \neq \emptyset$. Then Zorn's lemma implies that G has some $\sqsubseteq$ -maximal element s' . Then $s' \in S_{max}(\Sigma, i)$ and $s \sqsubseteq s'$.

Lemma is proved.

Lemma 6. Let $\Sigma = (T, Q, Tr)$ be a I/O NCMS, $Q' \subseteq Q$, and $i \in Sb(In(\Sigma), W)$. Then

- (1) $o_{all}(\Sigma, Q', i) \subseteq Sb(Out(\Sigma), W)$;
- (2) $dom(o) \subseteq dom(i)$ for each o in $o_{all}(\Sigma, Q', i)$;
- (3) $o_{all}(\Sigma, Q', i) \neq \emptyset$.

Proof.

(1) Let $o \in o_{all}(\Sigma, Q', i)$. If $Q' = \emptyset$ or $dom(i) \subseteq \{0\}$, then $o = \perp$ or o has a form $\{0\} \mapsto out(q)$ for some $q \in Q$, whence $o \in Sb(Out(\Sigma), W)$, because $out(q) \in Out(\Sigma)W$ for any $q \in Q$. Consider the case when $\{0\} \subset dom(i)$ and $Q' \neq \emptyset$. Then either o has a form $\{0\} \mapsto out(q)$ for some $q \in Q$, or $o = out \circ s$ for some $s \in S_{max}(\Sigma, i)$. In the former case, $o \in Sb(Out(\Sigma), W)$. In the latter case, $s \in Tr$, $dom(s) \in \mathbf{T}_0$, and $(out \circ s)(t) \in Out(\Sigma)W$ for all $t \in dom(s)$, whence $o = out \circ s \in Sb(Out(\Sigma), W)$. In all cases, $o \in Sb(Out(\Sigma), W)$.

(2) Let $o \in o_{all}(\Sigma, Q', i)$. If $Q' = \emptyset$ or $i = \perp$, then $o = \perp$ and $dom(o) \subseteq dom(i)$. If $Q' \neq \emptyset$ and $dom(i) = \{0\}$, then $dom(o) = \{0\} \subseteq dom(i)$. If $Q' \neq \emptyset$ and $\{0\} \subset dom(i)$, then either $dom(o) = \{0\} \subset dom(i)$, or $dom(o) = dom(out \circ s) = dom(s)$ for some $s \in S_{max}(\Sigma, i)$. In the latter case, $in \circ s \leq i$, whence $dom(o) = dom(s) \subseteq dom(i)$. In all cases, $dom(o) \subseteq dom(i)$.

(3) If $Q' = \emptyset$ or $i = \perp$ or $Q' \neq \emptyset$ and $i = \{0\}$, then $o_{all}(\Sigma, Q', i) \neq \emptyset$ immediately from definition of o_{all} . Consider the case when $Q' \neq \emptyset$ and $\{0\} \subset dom(i)$. If there exists $q \in Q' \setminus S_{init}(\Sigma, i)$, then $\{0\} \mapsto out(q)$ belongs to $o_{all}(\Sigma, Q', i)$. Otherwise, $Q' \subseteq S_{init}(\Sigma, i)$. Let us choose any $q \in Q'$ (it exists because $Q' \neq \emptyset$). Then $q \in S_{init}(\Sigma, i)$ and there exists $s \in S(\Sigma, i)$ such that $s(0) = q$. Then by Lemma 5, there exists $s' \in S_{max}(\Sigma, i)$ such that $s \sqsubseteq s'$, whence $s'(0) = q$. Then $out \circ s' \in o_{all}(\Sigma, Q', i)$. Lemma is proved.

Lemma 7. Each initial I/O NCMS is a NCMS representation of a unique (up to semantic identity) block.

Proof.

Uniqueness up to semantic identity is obvious. Let us prove that if (Σ, Q_0) is an initial I/O NCMS, where $\Sigma = (T, Q, Tr)$, then it is a NCMS representation of some block.

Let $i \in Sb(In(\Sigma), W)$. Let us show that $O_{all}(\Sigma, Q_0, i)$ is a non-empty subset of $Sb(Out(\Sigma), W)$ and $dom(o) \subseteq dom(i)$ for all $o \in O_{all}(\Sigma, Q_0, i)$. This is obvious, if $dom(i) = \emptyset$. Consider the case when $dom(i) \neq \emptyset$. For any $x \in IState(\Sigma)$ we have $Sel_{1,2}(Q_0, i(0), x) \subseteq Q_0 \subseteq Q$. Besides, $IState(\Sigma) \neq \emptyset$. Then Lemma 6 implies that $O_{all}(\Sigma, Q_0, i) \in 2^{Sb(Out(\Sigma), W) \setminus \{\emptyset\}}$ and $dom(o) \subseteq dom(i)$ for all $o \in O_{all}(\Sigma, Q_0, i)$. Thus (Σ, Q_0) is a NCMS representation of a block.

Lemma is proved.

Lemma 8. Let B be a deterministic causal block. Then B has a NCMS representation.

Proof.

Let $X = \{i \in Sb(In(B), W) \mid \exists t \in T \text{ } dom(i) = [0, t]\}$ and $Q = {}^{In(B)}W \times X \times {}^{Out(B)}W$. Then $X \neq \emptyset$. Let $in, istate, out$ denote projection maps from Q on the first, second, and third coordinate respectively. Let Tr be the set of all functions of the form $s : A \rightarrow Q$, where $A \in \mathbf{T}$, such that

(A) for each $t \in dom(s) \setminus \{0\}$ we have $dom(istate(s(t))) = [0, t]$ and $istate(s(t))(t) = in(s(t))$ and if $s(0) \downarrow$, then $istate(s(t))(0) = in(s(0))$;

(B) for each $t \in dom(s)$ we have $t \in dom(o)$ and $out(s(t)) = o(t)$, where o is a unique member of $Op(B)(istate(s(t)))$;

(C) if $t_1, t_2 \in dom(s) \setminus \{0\}$ and $t_1 \leq t_2$, then $istate(s(t_1)) \sqsubseteq istate(s(t_2))$.

Let us show that $\Sigma = (T, Q, Tr)$ is a NCMS.

Firstly, let us check that Tr is closed under proper restrictions. Let $s \in Tr$, $A \in \mathbf{T}$, and $A \subseteq dom(s)$. Then s satisfies (A), (B), (C). Then $dom(s|_A) = A \in \mathbf{T}$ and $s|_A$ satisfies (A), (B), (C), whence $s|_A \in Tr$.

Secondly, let us check that Tr is Markovian. Assume that $s_1, s_2 \in Tr$, $t^* = \max dom(s_1) = \min dom(s_2)$, and $s_1(t^*) = s_2(t^*)$. Let $s : dom(s_1) \cup dom(s_2) \rightarrow Q$ be a function such that $s(t) = s_1(t)$, if $t \in dom(s_1)$ and $s(t) = s_2(t)$, if $t \in dom(s_2)$.

Let us show that s satisfies (A). Let $t \in dom(s) \setminus \{0\}$. Then $t \in dom(s_1) \setminus \{0\}$ or $t \in dom(s_2) \setminus \{0\}$ and because s_1, s_2 satisfy (A), we have $dom(istate(s(t))) = [0, t]$ and $istate(s(t))(t) = in(s(t))$. Assume that $s(0) \downarrow$. Then $s_1(0) \downarrow$. If $t \in dom(s_1) \setminus \{0\}$, then $istate(s(t))(0) = istate(s_1(t))(0) = in(s_1(0)) = in(s(0))$, because s_1 satisfies (A). Otherwise, $t \in dom(s_2)$ and

$0 < t^* \leq t$. Because s_2 satisfies (C), we have $istate(s_2(t^*)) \sqsubseteq istate(s_2(t))$. Because $t^* \in dom(s_1) \setminus \{0\}$ and $s_1(0) \downarrow$, we have $istate(s_1(t^*))(0) = in(s_1(0))$. Then $istate(s(t))(0) = istate(s_2(t))(0) = istate(s_1(t^*))(0) = in(s_1(0)) = in(s(0))$. So s satisfies (A).

Moreover, s satisfies (B), because $s|_{dom(s_1)} = s_1$, $s|_{dom(s_2)} = s_2$, and s_1, s_2 satisfy (B). Let $t_1, t_2 \in dom(s) \setminus \{0\}$ and $t_1 \leq t_2$. If $t_1 \leq t^* \leq t_2$, then $istate(s(t_1)) \sqsubseteq istate(s(t^*)) \sqsubseteq istate(s(t_2))$, then $istate(s(t_1)) \sqsubseteq istate(s(t_2))$. Otherwise, both t_1, t_2 belong to $dom(s_1)$ or $dom(s_2)$ and $istate(s(t_1)) \sqsubseteq istate(s(t_2))$ also holds. Then s satisfies (C). Thus Tr is Markovian.

Thirdly, let us check that Tr is complete. Let $c \subseteq Tr$ be a non-empty $\sqsubseteq$ -chain. Let s^* be a function such that the graph of s^* is a union of graphs of all elements of c (this is indeed a function, because c is a chain). Then $dom(s^*) \in \mathbf{T}$ (because $c \neq \emptyset$) and s^* satisfies (A), (B), (C) because each $s \in c$ satisfies (A), (B), (C). Thus $s^* \in Tr$. It is easy to see that s^* is a least upper bound of c in Tr viewed as a poset with respect to $\sqsubseteq$. We conclude that Σ is a NCMS.

Let $i \in Sb(In(B), W)$ and $o \in Op(B)(i)$.

Let us show that for each $s \in S(\Sigma, i)$, $out \circ s = o|_{dom(s)}$, and if $s \in S_{max}(\Sigma, i)$, then $out \circ s = o$. Let $s \in S(\Sigma, i)$. Then $dom(s) \in \mathbf{T}_0$ and $in \circ s \leq i$ by definition of $S(\Sigma, i)$ and $istate(s(t))(t) = in(s(t)) = i(t)$ for all $t \in dom(s) \setminus \{0\}$ by (A). If $t', t \in dom(s)$ and $0 < t' \leq t$, then $i(t') = istate(s(t'))(t') = istate(s(t))(t')$ by (C). Besides, $s(0) \downarrow$, whence $istate(s(t))(0) = in(s(0)) = i(0)$ for each $t \in dom(s) \setminus \{0\}$ by (A). Then for each $t \in dom(s) \setminus \{0\}$ we have $istate(s(t)) = i|_{[0, t]}$, because $dom(istate(s(t))) = [0, t]$. Then $Op(B)(istate(s(t))) = Op(B)(i|_{[0, t]}) = \{o|_{[0, t]}\}$, because B is deterministic and causal. Then $out(s(t)) = (o|_{[0, t]})(t)$ and $t \in dom(o)$ for each $t \in dom(s)$ by (B). This implies that $dom(s) \subseteq dom(o)$ and for all $t \in dom(s)$, $out(s(t)) = o(t)$. Thus $out \circ s = o|_{dom(s)}$. We have $\{0\} \subset dom(s) \subseteq dom(o)$, so $dom(o) \in \mathbf{T}$. Because $in(s(0)) = i(0)$ and $out(s(0)) = o(0)$, it is easy to see that a function $s' : dom(o) \rightarrow Q$ such that $s'(0) = s(0)$ and $s'(t) = (i(t), i|_{[0, t]}, o(t))$ for all $t \in dom(o) \setminus \{0\}$ satisfies (A), (B), and (C). Moreover, $s' \in Tr$, $dom(s') \in \mathbf{T}_0$, and $in \circ s' = i|_{dom(o)} \leq i$. Then $s' \in S(\Sigma, i)$. Besides, $s'|_{dom(s)} = s$. This implies that if $s \in S_{max}(\Sigma, i)$, then $s' = s$ and $out \circ s = o$.

Let us denote $Q_0 = \{(d_{in}, x, d_{out}) \in Q \mid \exists (i, o) \in IO(B) \{0\} \in dom(o) \wedge d_{in} = i(0) \wedge d_{out} = o(0)\}$.

Let us prove $S_{init}(\Sigma) \subseteq Q_0$. Let $(d_{in}, x, d_{out}) \in S_{init}(\Sigma)$. Then $(d_{in}, x, d_{out}) = s(0)$ for some $s \in Tr$ such

that $dom(s) \in T_0$, then $s \in S(\Sigma, in \circ s)$, whence $out \circ s = o|_{dom(s)}$, where o is a unique member of $Op(B)(in \circ s)$. Then $(in \circ s, o) \in IO(B)$, $\{0\} \in dom(o)$, and $d_{in}=in(s(0))=(in \circ s)(0)$, and $d_{out}=out(s(0))=o(0)$. Thus $(d_{in}, x, d_{out}) \in Q_0$.

We conclude that (Σ, Q_0) is an initial I/O NCMS. Obviously, $In(\Sigma)=In(B)$ and $Out(\Sigma)=Out(B)$.

Let us prove the following property of Q_0 :

(D) if $(i, o) \in IO(B)$, $q \in Q_0$, $i \neq \perp$, and $in(q)=i(0)$, then $o \neq \perp$ and $out(q)=o(0)$.

Indeed, if $(i, o) \in IO(B)$, $i \neq \perp$, and $q \in Q_0$, then there exists $(i', o') \in IO(B)$ such that $\{0\} \in dom(o') \subseteq dom(i')$, $in(q)=i'(0)$, $out(q)=o'(0)$. Because $i'|_{\{0\}}=i|_{\{0\}}$ and B is deterministic and causal, we have $o'|_{\{0\}}=o|_{\{0\}}$, whence $o \neq \perp$ and $o(0)=o'(0)=out(q)$.

Now let us show that (Σ, Q_0) is a NCMS representation of B .

It is sufficient to show that $Op(B)(i)=O_{all}(\Sigma, Q_0, i)$ for all $i \in Sb(In(B), W)$. This is obvious, if $i=\perp$.

Let $i \in Sb(In(B), W) \setminus \{\perp\}$ and $o \in Op(B)(i)$. Consider the following cases:

1) $Sel_{1,2}(Q_0, i(0), x) = \emptyset$ for some $x \in IState(\Sigma)$. Then there is no $(i', o') \in IO(B)$ such that $i'(0)=i(0)$ and $o'(0) \downarrow$. Then $o=\perp$ and $Sel_{1,2}(Q_0, i(0), x) = \emptyset$ for all $x \in IState(\Sigma)$. Then $O_{all}(\Sigma, Q_0, i)=\{\perp\}=Op(B)(i)$.

2) $Sel_{1,2}(Q_0, i(0), x) \neq \emptyset$ for all $x \in IState(\Sigma)$ and $dom(i)=\{0\}$. Then $o(0) \downarrow$ and $out(q)=o(0)$ for each $q \in Sel_{1,2}(Q_0, i(0), x) \subseteq Q_0$ by the property (D). Then $o_{all}(\Sigma, Sel_{1,2}(Q_0, i(0), x), i)=\{\{0\} \mapsto o(0)\}=\{o\}$ for all $x \in IState(\Sigma)$, whence $O_{all}(\Sigma, Q_0, i)=Op(B)(i)$.

3) $Sel_{1,2}(Q_0, i(0), x) \neq \emptyset$ for all $x \in IState(\Sigma)$, $\{0\} \subset dom(i)$, and $dom(o) \subseteq \{0\}$. If $in(q)=i(0)$ for some $q \in S_{init}(\Sigma, i)$, then $q=s(0)$ for some $s \in S(\Sigma, i)$, whence $out \circ s = o|_{dom(s)}$ as we have shown above, but this is impossible, because $\{0\} \subset dom(s)$ and $dom(o) \subseteq \{0\}$. Thus $in(q) \neq i(0)$ for each $q \in S_{init}(\Sigma, i)$. Then for each $x \in IState(\Sigma)$, $s(0) \notin Sel_{1,2}(Q_0, i(0), x)$ for all $s \in S_{max}(\Sigma, i)$ and $Sel_{1,2}(Q_0, i(0), x) \cap S_{init}(\Sigma, i)=\emptyset$. Then $o_{all}(\Sigma, Sel_{1,2}(Q_0, i(0), x), i) = \{\{0\} \mapsto out(q) \mid q \in Sel_{1,2}(Q_0, i(0), x)\}$ for each $x \in IState(\Sigma)$, whence $O_{all}(\Sigma, Q_0, i)=\{\{0\} \mapsto out(q) \mid q \in Q_0 \wedge in(q)=i(0)\} \neq \emptyset$. Because $in(q)=i(0)$ for some $q \in Q_0$, by the property (D) we have $0 \in dom(o)$ and $O_{all}(\Sigma, Q_0, i)=\{\{0\} \mapsto o(0)\}=\{o\}=Op(B)(i)$.

4) $Sel_{1,2}(Q_0, i(0), x) \neq \emptyset$ for all $x \in IState(\Sigma)$ and $\{0\} \subset dom(o)$. We have $dom(o) \in T$. Let $x \in IState(\Sigma)$ and $q \in Sel_{1,2}(Q_0, i(0), x)$. Then $in(q)=i(0)$ and have $out(q)=o(0)$ by the property (D). It is easy to see that a function $s' : dom(o) \rightarrow Q$ such that

$s'(0)=q$ and $s'(t) = (i(t), i|_{[0,t]}, o(t))$ for all $t \in dom(o) \setminus \{0\}$ satisfies (A), (B), and (C). Moreover, $s' \in Tr$, $dom(s') \in T_0$, and $in \circ s' = i|_{dom(o)} \leq i$. Then $s' \in S(\Sigma, i)$. Then $s'(0)=q \in S_{init}(\Sigma, i)$. Because $q \in Sel_{1,2}(Q_0, i(0), x)$ is arbitrary, we have $Sel_{1,2}(Q_0, i(0), x) \subseteq S_{init}(\Sigma, i)$. Then $o_{all}(\Sigma, Sel_{1,2}(Q_0, i(0), x), i) = \{out \circ s \mid s \in S_{max}(\Sigma, i) \wedge s(0) \in Sel_{1,2}(Q_0, i(0), x)\} = \{o\}$ for each $x \in IState(\Sigma)$, because $out \circ s = o$ for any $s \in S_{max}(\Sigma, i)$ as we have show above and $Sel_{1,2}(Q_0, i(0), x) \neq \emptyset$. Then $O_{all}(\Sigma, Q_0, i) = \{o\} = Op(B)(i)$, because $IState(\Sigma) \neq \emptyset$.

In all cases 1-4 we have $O_{all}(\Sigma, Q_0, i)=Op(B)(i)$. Thus (Σ, Q_0) is a NCMS representation of the block B . Lemma is proved.

Let $\Sigma_1=(T, Q_1, Tr_1)$ and $\Sigma_2=(T, Q_2, Tr_2)$ be I/O NCMS such that $In(\Sigma_1)=In(\Sigma_2)$ and $Out(\Sigma_1)=Out(\Sigma_2)$.

Let us introduce the following notions.

Definition 17.

(1) A state embedding from Σ_1 to Σ_2 is a function

$f: Q_1 \rightarrow Q_2$ such that $\{f \circ s \mid s \in Tr_1\} = \{s \in Tr_2 \mid \exists t \in dom(s) \exists q \in Q_1 s(t)=f(q)\}$ and there exists an injective function $g: IState(\Sigma_1) \rightarrow IState(\Sigma_2)$ such that $f(q)=(in(q), g(istate(q)), out(q))$, $q \in Q_1$.

(2) A state embedding from an initial I/O NCMS (Σ_1, Q_0^1) to an initial I/O NCMS (Σ_2, Q_0^2) is a state embedding f from Σ_1 to Σ_2 such that for each $q \in Q_1$, $q \in Q_0^1$ iff $f(q) \in Q_0^2$.

It follows immediately that a state embedding from Σ_1 to Σ_2 is an injective function.

Lemma 9. Let $\Sigma_1=(T, Q_1, Tr_1)$ and $\Sigma_2=(T, Q_2, Tr_2)$ be I/O NCMS, $In(\Sigma_1)=In(\Sigma_2)$ and $Out(\Sigma_1)=Out(\Sigma_2)$, and f be a state embedding from Σ_1 to Σ_2 .

Let $i \in Sb(In(\Sigma_1), W)$. Then $S_{max}(\Sigma_2, i) \supseteq \{f \circ s \mid s \in S_{max}(\Sigma_1, i)\}$ and $\{q \in S_{init}(\Sigma_2, i) \mid \exists q' \in Q_1 q=f(q')\} = \{f(q'') \mid q'' \in S_{init}(\Sigma_1, i)\}$.

Proof.

Because $\{f \circ s \mid s \in Tr_1\} \subseteq Tr_2$, we have $S(\Sigma_2, i) = \{s \in Tr_2 \mid dom(s) \in T_0 \wedge in \circ s \leq i\} \supseteq \{f \circ s \mid s \in Tr_1 \wedge dom(f \circ s) \in T_0 \wedge in \circ (f \circ s) \leq i\} = \{f \circ s \mid s \in Tr_1 \wedge dom(s) \in T_0 \wedge in \circ s \leq i\} = \{f \circ s \mid s \in S(\Sigma_1, i)\}$.

Let $s \in S_{max}(\Sigma_1, i)$. Then $f \circ s \in S(\Sigma_2, i)$. Suppose that $f \circ s \notin S_{max}(\Sigma_2, i)$. Then $f \circ s \sqsubset s'$ for some $s' \in S(\Sigma_2, i)$. Because $dom(s) \neq \emptyset$, $s'(t) \in \{f(q) \mid q \in Q_1\}$ for some $t \in T$. Then $s' = f \circ s''$ for some $s'' \in Tr_1$, because $\{f \circ s \mid s \in Tr_1\} = \{s \in Tr_2 \mid \exists t \in dom(s) \exists q \in Q_1 s(t)=f(q)\}$. Then $f \circ s \sqsubset f \circ s''$, whence $s \sqsubset s''$, because f is injective. Besides, $dom(s'')=dom(s') \in T_0$

and $in \circ s'' = in \circ s' \leq i$. Then $s'' \in S(\Sigma_1, i)$. We get a contradiction with the assumption $s \in S_{\max}(\Sigma_1, i)$. We conclude that $f \circ s \in S_{\max}(\Sigma_2, i)$.

Thus we have $S_{\max}(\Sigma_2, i) \supseteq \{f \circ s \mid s \in S_{\max}(\Sigma_1, i)\}$.

Now let us show that $\{q \in S_{\text{init}}(\Sigma_2, i) \mid \exists q' \in Q_1 \ q = f(q')\} = \{f(q'') \mid q'' \in S_{\text{init}}(\Sigma_1, i)\}$. Let $q \in S_{\text{init}}(\Sigma_2, i)$ and $q' \in Q_1$ be such that $q = f(q')$. Then $q = s(0)$ for some $s \in S(\Sigma_2, i)$. Then $s \in Tr_2$ and $s(0) = f(q')$, where $q' \in Q_1$. Then there exists $s' \in Tr_1$ such that $s = f \circ s'$. Moreover, $dom(s') = dom(s) \in T_0$ and $in \circ s' = in \circ s \leq i$. Then $s' \in S(\Sigma_1, i)$ and $s'(0) \in S_{\text{init}}(\Sigma_1, i)$. Thus $q = s(0) = f(s'(0)) \in \{f(q'') \mid q'' \in S_{\text{init}}(\Sigma_1, i)\}$.

Conversely, let $q'' \in S_{\text{init}}(\Sigma_1, i)$. Then $q'' = s(0)$ for some $s \in S(\Sigma_1, i)$. Then $f \circ s \in S(\Sigma_2, i)$, whence $f(q'') = f(s(0)) = (f \circ s)(0) \in S_{\text{init}}(\Sigma_2, i)$. Then because $q'' \in Q_1$, we have $f(q'') \in \{q \in S_{\text{init}}(\Sigma_2, i) \mid \exists q' \in Q_1 \ q = f(q')\}$.

Thus $\{q \in S_{\text{init}}(\Sigma_2, i) \mid \exists q' \in Q_1 \ q = f(q')\} = \{f(q'') \mid q'' \in S_{\text{init}}(\Sigma_1, i)\}$. Lemma is proved.

Lemma 10. For $j=1,2$ let (Σ_j, Q_0^j) be a NCMS representation of a block B_j . Assume $In(\Sigma_1) = In(\Sigma_2)$, $Out(\Sigma_1) = Out(\Sigma_2)$, and there exists a state embedding f from (Σ_1, Q_0^1) to (Σ_2, Q_0^2) . Then $B_1 \trianglelefteq B_2$.

Proof.

Assume that $\Sigma_1 = (T, Q_1, Tr_1)$ and $\Sigma_2 = (T, Q_2, Tr_2)$.

We have $In(B_1) = In(\Sigma_1) = In(\Sigma_2) = In(B_2)$ and $Out(B_1) = Out(\Sigma_1) = Out(\Sigma_2) = Out(B_2)$. Because f is a state embedding, there exists an injective function $g : IState(\Sigma_1) \rightarrow IState(\Sigma_2)$ such that $f(q) = (in(q), g(istate(q)), out(q))$ for all $q \in Q$.

Let $i \in Sb(In(B), W)$. Then for $j=1,2$, $Op(B_j)(i) = O_{all}(\Sigma_j, Q_0^j, i)$. Let us show that $O_{all}(\Sigma_2, Q_0^2, i) \supseteq O_{all}(\Sigma_1, Q_0^1, i)$. This is obvious, if $i \perp$, so assume that $i \neq \perp$. Let us fix some $x_1 \in IState(\Sigma_1)$. Denote $Q_1' = Sel_{1,2}(Q_0^1, i(0), x_1)$ and $Q_2' = Sel_{1,2}(Q_0^2, i(0), g(x_1))$. Because g is injective and $Q_0^2 \supseteq \{f(q) \mid q \in Q_0^1\}$, $Q_2' = \{q \in Q_0^2 \mid in(q) = i(0) \wedge istate(q) = g(x_1)\} \supseteq \{f(q) \mid q \in Q_0^1 \wedge in(f(q)) = i(0) \wedge istate(f(q)) = g(x_1)\} = \{f(q) \mid q \in Q_0^1 \wedge in(q) = i(0) \wedge g(istate(q)) = g(x_1)\} = \{f(q) \mid q \in Sel_{1,2}(Q_0^1, i(0), x_1)\} = \{f(q) \mid q \in Q_1'\}$.

Let us show that $Q_2' \neq \emptyset$ iff $Q_1' \neq \emptyset$.

Indeed, if $Q_1' \neq \emptyset$, then $Q_2' \neq \emptyset$, because $Q_2' \supseteq \{f(q) \mid q \in Q_1'\}$. Conversely, if $Q_2' \neq \emptyset$, then $(i(0), g(x_1), d) \in Q_0^2$ for some $d \in Out(\Sigma_2)$, whence $(i(0), x_1, d) \in Q_1$ and $f(i(0), x_1, d) \in Q_0^2$, whence, $(i(0), x_1, d) \in Q_0^1$, because f is a state embedding, and finally, $Q_1' = Sel_{1,2}(Q_0^1, i(0), x_1) \neq \emptyset$.

Now let us show that $O_{all}(\Sigma_2, Q_2', i) \supseteq O_{all}(\Sigma_1, Q_1', i)$. This is obvious, if $Q_1' = \emptyset$ or $Q_2' = \emptyset$, because $Q_2' \neq$

$\emptyset$ iff $Q_1' \neq \emptyset$ as we have shown above. So assume that $Q_1' \neq \emptyset$ and $Q_2' \neq \emptyset$.

Consider the case when $dom(i) = \{0\}$. Because $Q_2' \supseteq \{f(q) \mid q \in Q_1'\}$, we have $O_{all}(\Sigma_2, Q_2', i) = \{\{0\} \mapsto out(q) \mid q \in Q_2'\} \supseteq \{\{0\} \mapsto out(f(q)) \mid q \in Q_1'\} = \{\{0\} \mapsto out(q) \mid q \in Q_1'\} = O_{all}(\Sigma_1, Q_1', i)$.

Now consider the case when $\{0\} \subset dom(i)$. Then $O_{all}(\Sigma_2, Q_2', i) = \{out \circ s \mid s \in S_{\max}(\Sigma_2, i) \wedge s(0) \in Q_2'\} \cup \{\{0\} \mapsto out(q) \mid q \in Q_2' \setminus S_{\text{init}}(\Sigma_2, i)\}$. By Lemma 9 we have $S_{\max}(\Sigma_2, i) \supseteq \{f \circ s \mid s \in S_{\max}(\Sigma_1, i)\}$ and $\{q \in S_{\text{init}}(\Sigma_2, i) \mid \exists q' \in Q_1 \ q = f(q')\} = \{f(q'') \mid q'' \in S_{\text{init}}(\Sigma_1, i)\}$. Because f is injective and $Q_2' \supseteq \{f(q) \mid q \in Q_1'\}$, $\{out \circ s \mid s \in S_{\max}(\Sigma_2, i) \wedge s(0) \in Q_2'\} \supseteq \{out \circ (f \circ s) \mid s \in S_{\max}(\Sigma_1, i) \wedge f(s(0)) \in Q_2'\} \supseteq \{out \circ s \mid s \in S_{\max}(\Sigma_1, i) \wedge s(0) \in Q_1'\}$. Moreover, because $Q_1' \subseteq Q_1$ and f is injective, we have $Q_2' \setminus S_{\text{init}}(\Sigma_2, i) \supseteq \{f(q) \mid q \in Q_1' \setminus S_{\text{init}}(\Sigma_2, i)\} = \{f(q) \mid q \in Q_1'\} \setminus \{f(q'') \mid q'' \in S_{\text{init}}(\Sigma_1, i)\} = \{f(q) \mid q \in Q_1' \setminus S_{\text{init}}(\Sigma_1, i)\}$. Then $\{\{0\} \mapsto out(q) \mid q \in Q_2' \setminus S_{\text{init}}(\Sigma_2, i)\} \supseteq \{\{0\} \mapsto out(f(q)) \mid q \in Q_1' \setminus S_{\text{init}}(\Sigma_1, i)\} = \{\{0\} \mapsto out(q) \mid q \in Q_1' \setminus S_{\text{init}}(\Sigma_1, i)\}$. Finally, $O_{all}(\Sigma_2, Q_2', i) \supseteq O_{all}(\Sigma_1, Q_1', i)$.

We conclude that for each $x_1 \in IState(\Sigma_1)$, $O_{all}(\Sigma_2, Sel_{1,2}(Q_0^2, i(0), g(x_1)), i)$ is a superset of $O_{all}(\Sigma_1, Sel_{1,2}(Q_0^1, i(0), x_1), i)$.

Then because $i \neq \perp$ by our assumption, we have $Op(B_2)(i) = O_{all}(\Sigma_2, Q_0^2, i) \supseteq O_{all}(\Sigma_1, Q_0^1, i) = Op(B_1)(i)$.

We conclude that $B_1 \trianglelefteq B_2$. Lemma is proved.

Definition 18. A disjoint union of an indexed family of initial I/O NCMS $((\Sigma_j, Q_0^j))_{j \in J}$, where $J \neq \emptyset$ and $\Sigma_j = (T, Q_j, Tr_j)$ for each $j \in J$, is a pair (Σ, Q_0) , where $\Sigma = (T, Q, Tr)$ and $Q_0 = \{f_j(q) \mid j \in J \wedge q \in Q_0^j\}$, where $Q = {}^{IN}W \times (\bigcup_{j \in J} \{j\} \times IState(\Sigma_j)) \times {}^{OUT}W$,

$In = \bigcup_{j \in J} In(\Sigma_j)$, $Out = \bigcup_{j \in J} Out(\Sigma_j)$, and

$Tr = \{f_j \circ s \mid j \in J \wedge s \in Tr_j\}$, and

for each $j \in J$, $f_j : Q_j \rightarrow Q$ is a function such that $f_j(q) = (in(q), (j, istate(q)), out(q))$, $q \in Q_j$.

Lemma 11. Let (Σ, Q_0) be a disjoint union of an indexed family of initial I/O NCMS $((\Sigma_j, Q_0^j))_{j \in J}$, where $J \neq \emptyset$. Then (Σ, Q_0) is an initial I/O NCMS.

Proof.

Assume $\Sigma = (T, Q, Tr)$ and $\Sigma_j = (T, Q_j, Tr_j)$ for each $j \in J$. Let In , Out , and $f_j : Q_j \rightarrow Q$, $j \in J$ be defined as in Definition 18. Let us show that Σ is a NCMS. For each $j \in J$ let $Q_j' = {}^{IN}W \times (\{j\} \times IState(\Sigma_j)) \times {}^{OUT}W$. Then f_j is an injective function from Q_j to Q_j' .

Then because Σ_j is a NCMS, the triple (T, Q_j, Tr_j) , where $Tr_j = \{f_j \circ s \mid s \in Tr_j\}$, is a NCMS by Lemma 3.

For each $j, j' \in J$ such that $j \neq j'$ we have $Q'_j \cap Q'_{j'} = \emptyset$. Moreover, $Q = \bigcup_{j \in J} Q_j$ and $Tr = \bigcup_{j \in J} Tr_j$.

by Definition 18. Then $\Sigma = (T, Q, Tr)$ is a NCMS by Lemma 4. Because $J \neq \emptyset$, Σ is an I/O NCMS.

For each $j \in J$, Σ_j is an initial I/O NCMS, so $S_{init}(\Sigma_j) \subseteq Q_0^j \subseteq Q_j$. Then

$$\begin{aligned} S_{init}(\Sigma) &= \{s(0) \mid s \in Tr \wedge \text{dom}(s) \in T_0\} = \\ &= \{(f_j \circ s)(0) \mid j \in J \wedge s \in Tr_j \wedge \text{dom}(f_j \circ s) \in T_0\} = \\ &= \{f_j(q) \mid j \in J \wedge q \in S_{init}(\Sigma_j)\} \subseteq \\ &\subseteq \{f_j(q) \mid j \in J \wedge q \in Q_0^j\} = Q_0. \end{aligned}$$

Obviously, $Q_0 \subseteq Q$, so $S_{init}(\Sigma) \subseteq Q_0 \subseteq Q$. Thus (Σ, Q_0) is an initial I/O NCMS. Lemma is proved.

Definition 19.

- (1) A complete set of sub-blocks of a block B is a set C of sub-blocks of B such that $IO(B) = \bigcup_{B' \in C} IO(B')$
- (2) A complete indexed family of sub-blocks of a block B is an indexed family $(B_j)_{j \in J}$ such that $\{B_j \mid j \in J\}$ is a complete set of sub-blocks of B .

Lemma 12. Let $(B_j)_{j \in J}$ be a complete indexed family of sub-blocks of a block B , where $J \neq \emptyset$. Assume that for each $j \in J$, B_j has a NCMS representation (Σ_j, Q_0^j) . Let (Σ, Q_0) be a disjoint union of $((\Sigma_j, Q_0^j))_{j \in J}$. Then (Σ, Q_0) is a NCMS representation of B .

Proof.

Assume that $\Sigma_j = (T, Q_j, Tr_j)$ for each $j \in J$ and $\Sigma = (T, Q, Tr)$. By Lemma 11, (Σ, Q_0) is an initial I/O NCMS, whence by Lemma 7, there exists a block B' (unique up to semantic identity) such that (Σ, Q_0) is a NCMS representation of B' .

For each $j \in J$ we have $In(\Sigma_j) = In(B_j) = In(B)$ and $Out(\Sigma_j) = Out(B_j) = Out(B)$, because $B_j \trianglelefteq B$. Because $J \neq \emptyset$, $In(B') = In(\Sigma) = In(B)$ and $Out(B') = Out(\Sigma) = Out(B)$. For each $j \in J$, let $g_j : IState(\Sigma_j) \rightarrow IState(\Sigma)$ and $f_j : Q_j \rightarrow Q$ be functions such that $g_j(x) = (j, x)$ for all $x \in IState(\Sigma_j)$ and $f_j(q) = (in(q), g_j(istate(q)), out(q))$ for all $q \in Q_j$.

Let us show that $B \trianglelefteq B'$. Let $j \in J$. Obviously, $In(\Sigma_j) = In(\Sigma)$, $Out(\Sigma_j) = Out(\Sigma)$, and g_j is injective. Because $\{f_j(q) \mid q \in Q_j\} \cap \{f_{j'}(q) \mid q \in Q_{j'}\} = \emptyset$, if $j', j'' \in J$ and $j' \neq j''$, from the item (2) of Definition 18 it follows that $\{f_j \circ s \mid s \in Tr_j\} = \{s \in Tr \mid \exists t \in \text{dom}(s) \exists q \in Q_j s(t) = f_j(q)\}$. Then f_j is a state embedding from Σ_j to Σ . Moreover, from the item (3) of Definition 18 it follows that for each $q \in Q_j$, $q \in$

Q_0^j iff $f_j(q) \in Q_0$, because f_j is injective and $\{f_j(q) \mid q \in Q_j\} \cap \{f_{j'}(q) \mid q \in Q_{j'}\} = \emptyset$ for all $j', j'' \in J$ such that $j' \neq j''$. Thus f_j is a state embedding from (Σ_j, Q_0^j) to (Σ, Q_0) . Then $B_j \trianglelefteq B'$ by Lemma 10. Because $j \in J$ is arbitrary, we conclude that $B \trianglelefteq B'$.

Now let us show that $B' \trianglelefteq B$. Let $(i, o) \in IO(B')$. Then $o \in O_{all}(\Sigma, Q_0, i)$. If $i = \perp$, then $o = \perp$ and $(i, o) \in IO(B)$ (because B is a block). Consider the case when $i \neq \perp$. Then there exists $x^* \in IState(\Sigma)$ such that $o \in o_{all}(\Sigma, Sel_{1,2}(Q_0, i(0), x^*), i)$. Then there exists $j \in J$ and $x_j^* \in IState(\Sigma_j)$ such that $x^* = (j, x_j^*)$. Let $Q'_j = {}^{In(\Sigma)}W \times (\{j\} \times IState(\Sigma_j)) \times {}^{Out(\Sigma)}W$, and Σ'_j is the restriction of Σ on Q'_j . By Lemma 1, Σ'_j is a NCMS. We will denote by Tr'_j the set of trajectories of Σ'_j . Moreover, Q'_j is the set of states of Σ'_j and $In(\Sigma'_j) = In(\Sigma)$, $Out(\Sigma'_j) = Out(\Sigma)$. Besides, Σ'_j is an I/O NCMS and $S_{init}(\Sigma'_j) \subseteq S_{init}(\Sigma) \cap Q'_j \subseteq Q_0 \cap Q'_j \subseteq Q'_j$, because (Σ, Q_0) is an initial I/O NCMS. Denote $Q_{0,j}' = Q_0 \cap Q'_j$. Then $(\Sigma'_j, Q_{0,j}')$ is an initial I/O NCMS. Moreover, $x^* \in IState(\Sigma'_j)$.

Let us prove that $o \in O_{all}(\Sigma'_j, Q'_{0,j}, i)$. Denote $Q' = Sel_{1,2}(Q_{0,j}', i(0), x^*)$. Firstly, let us show that $Sel_{1,2}(Q_{0,j}', i(0), x^*) = Q'$. Indeed, if $q \in Q'_{0,j} = Q_0 \cap Q'_j$ and $in(q) = i(0)$, $istate(q) = x^*$, then $q \in Q'$. Conversely, if $q \in Q'$, then $q \in Q_0$, $in(q) = i(0)$, $istate(q) = (j, x_j^*)$, whence $q \in Q'_j$ and $q \in Sel_{1,2}(Q_{0,j}', i(0), x^*)$.

Secondly, let us show that $o \in o_{all}(\Sigma'_j, Q', i)$. Note that $o \in o_{all}(\Sigma, Q', i)$. If $Q' = \emptyset$ or $\text{dom}(i) \subseteq \{0\}$, from the definition of o_{all} we have $o \in o_{all}(\Sigma, Q', i) = o_{all}(\Sigma'_j, Q', i)$. Consider the case when $Q' \neq \emptyset$ and $\{0\} \subset \text{dom}(i)$. Because $o \in o_{all}(\Sigma, Q', i)$, the following two sub-cases are possible:

1) $o = out \circ s$ for some $s \in S_{max}(\Sigma, i)$ such that $s(0) \in Q'$. Then $s \in Tr$, $\text{dom}(s) \in T_0$, and $in \circ s \leq i$. Because $s(0) \in Q'$, we have $istate(s(0)) = x^* = (j, x_j^*)$. From the item (2) of Definition 18 for Σ it follows that the first component of the value $istate(s(t))$ is j for all $t \in \text{dom}(s)$. Then $s(t) \in Q'_j$ for all $t \in \text{dom}(s)$, whence $s \in Tr'_j$. Then $s \in S(\Sigma'_j, i)$. Moreover, $s \in S_{max}(\Sigma'_j, i)$, because otherwise, $s \sqsubset s'$ for some $s' \in S(\Sigma'_j, i)$, whence $s' \in S(\Sigma, i)$, and we get a contradiction with $s \in S_{max}(\Sigma, i)$. Thus $o \in o_{all}(\Sigma'_j, Q', i)$ by definition of o_{all} .

2) $o = \{0\} \mapsto out(q)$ for some $q \in Q' \setminus S_{init}(\Sigma, i)$. Then $q \neq s(0)$ for all $s \in Tr$ such that $\text{dom}(s) \in T_0$ and $in \circ s \leq i$. Because Σ'_j is a restriction of Σ on Q'_j , we have, in particular, $q \neq s(0)$ for all $s \in Tr'_j$ such that $\text{dom}(s) \in T_0$ and $in \circ s \leq i$. Then $q \in Q' \setminus S_{init}(\Sigma'_j, i)$, whence $o \in o_{all}(\Sigma'_j, Q', i)$.

Thus we conclude that $o \in o_{all}(\Sigma_{j'}, Q', i)$, $Q' = Sel_{1,2}(Q_{0,j'}, i(0), x^*)$, and $x^* \in IState(\Sigma_{j'})$, whence $o \in O_{all}(\Sigma_{j'}, Q_{0,j'}, i)$, because $i \neq \perp$ by assumption.

By Lemma 7, there exists a block $B'_{j'}$ such that $(\Sigma_{j'}, Q_{0,j'})$ is a NCMS representation of $B'_{j'}$. Let $g : IState(\Sigma_{j'}) \rightarrow IState(\Sigma_j)$ and $f : Q_{j'} \rightarrow Q_j$ be functions such that $g((j, x)) = x$ for all $x \in IState(\Sigma_{j'})$ and $f(q) = (in(q), g(istate(q)), out(q))$ for all $q \in Q_{j'}$. Obviously, $In(\Sigma_{j'}) = In(\Sigma_j)$, $Out(\Sigma_{j'}) = Out(\Sigma_j)$, and g is injective. Moreover, f is an inverse of f_j , whence

$\{f \circ s \mid s \in Tr_{j'}\} = \{f \circ s \mid s \in Tr \wedge \forall t \in dom(s) s(t) \in Q_{j'}\} = \{f \circ (f_j \circ s) \mid j' \in J \wedge s \in Tr' \wedge \forall t \in dom(s) (f_j \circ s)(t) \in Q_{j'}\} = Tr_j$. Because for any $s \in Tr_j$, $dom(s) \neq \emptyset$ and for each $t \in dom(s)$, $s(t) = f(f_j(s(t)))$, where $f_j(s(t)) \in Q_{j'}$, we have $\{f \circ s \mid s \in Tr_{j'}\} = \{s \in Tr_j \mid \exists t \in dom(s) \exists q \in Q_{j'} s(t) = f(q)\}$. Then f is a state embedding from $\Sigma_{j'}$ to Σ_j . Moreover, for each $q \in Q_{j'}$, $q \in Q_{0,j'} = Q_0 \cap Q_{j'}$ iff $q = f(q')$ for some $q' \in Q_0^j$ iff $f(q) \in Q_0^j$. Then f is a state embedding from $(\Sigma_{j'}, Q_{0,j'})$ to (Σ_j, Q_0^j) . Then $B_{j'} \leq B_j$ by Lemma 10. As we have shown above, $o \in O_{all}(\Sigma_{j'}, Q'_{0,j'}, i) = Op(B_{j'})(i)$, so $o \in Op(B_j)(i)$, whence $(i, o) \in IO(B)$.

We conclude that $B' \leq B$. We have shown that $B \leq B'$ and $B' \leq B$, so (Σ, Q_0) is a NCMS representation of B . Lemma is proved.

Now we can prove Theorem 1.

Proof of Theorem 1.

Let B be a strongly nonanticipative block. Let us show that B has a NCMS representation.

Let Rel be the set of all relations $R \subseteq IO(B)$ such that R is an I/O relation of a deterministic causal block. For each $R \in Rel$ let us define a block B_R such that $IO(B_R) = R$, $In(B_R) = In(B)$, $Out(B_R) = Out(B)$. Then B_R is a deterministic causal block for each $R \in Rel$ and $IO(B) = \bigcup_{R \in Rel} IO(B_R)$, because B is strongly nonanticipative. Then $(B_R)_{R \in Rel}$ is a complete indexed family of sub-blocks of B and $Rel \neq \emptyset$. By Lemma 8, for each $R \in Rel$ there exists an initial I/O NCMS (Σ_R, Q_0^R) which is a NCMS representation of B_R . Let (Σ, Q_0) be a disjoint union of $((\Sigma_R, Q_0^R))_{R \in Rel}$. Then by Lemma 12, (Σ, Q_0) is a NCMS representation of B .

Theorem is proved.

References

1. Campbell S.L., Chancelier J.-P., Nikoukhah R. Modeling and Simulation in Scilab/Scicos with ScicosLab 4.4. – Springer, 2010. – 342 p.
2. Sifakis J. A vision for computer science – the system perspective // Cent. Eur. J. Comp. Sci. – No. 1. – 2011. – P. 108-116.
3. Zadeh L.A., Desoer, C.A. Linear System Theory: The State Space Approach. – McGraw-Hill, 1963. – 628 p.
4. Mesarovic M.D., Takahara Y. Abstract Systems Theory – Springer, 1989. – 439 p.
5. Zeigler B.P., Praehofer H., Kim T.G. Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems. – Academic press, 2000. – 510 p.
6. Ivanov Ie. An Abstract Block Formalism for Engineering Systems // 2nd International Workshop on Algebraic, Logical, and Algorithmic Methods of System Modeling, Specification and Verification (SMSV 2013), June 18-22, 2013, Kherson, Ukraine. – 2013. – P. 448-463.
7. Ivanov Ie. A criterion for existence of global-in-time trajectories of non-deterministic Markovian systems // Communications in Computer and Information Science (CCIS) – Vol. 347. – 2012. – P. 111-130.
8. Nikitchenko N.S. A composition nominative approach to program semantics. Technical report IT-TR 1998-020, Technical University of Denmark, 1998. – 103 p.
9. Hájek O. Theory of processes I // Czechoslovak Mathematical Journal. – Vol. 17. – 1967. – P. 159-199.
10. Hájek O. Theory of processes II // Czechoslovak Mathematical Journal. – Vol. 17. – 1967. – P. 372-398.
11. Matrosov V.M., Anapolskiy L.Yu, Vasilyev, S.N. The method of comparison in mathematical systems theory. – Novosibirsk: Nauka, 1980. – 480 p. (in Russian)

Надійшла до редколегії 27.06.2013

УДК 519.876:517.6

T.V. Kolianova¹, assistant

Influence of delay on character of stability of stationary points for mathematical model of purchased immunity

A mathematical model of purchased immunity, which is described by a system of nonlinear differential equations with delay. Influence of delay is analyzed on stability of stationary points, and in the end, we give the biological consequences.

Keywords: system of nonlinear differential equations, the immune system, antigen, the stability of the stationary point.

¹ Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova st., 4d,

e-mail: tania.kolianova@gmail.com

Статтю представив д. т. н., с. н. с. Кудін В.І.

To study the effect of reaction delay lymphocytes to change amount of antigen we will consider a nonlinear system of ordinary differential equations describing the effect of purchased immunity:

$$\begin{aligned} \frac{dN}{dt} &= N(\mu_N - \gamma_1 L - \gamma_2 X), \\ \frac{dL}{dt} &= (L^* - L)(\mu_L - \gamma_3 N(t - \tau) - \gamma_4 G). \end{aligned} \quad (1)$$

The initial conditions are given by

$$\begin{aligned} N(t_0) &= N_0, L(t_0) = L_0, \\ N(t) &\equiv 0 \text{ for } t_0 - \tau < t < t_0. \end{aligned} \quad (2)$$

Suppose the parameter τ small. Replace function $N(t - \tau)$ its linear approximation:

$$N(t - \tau) \cong N(t) - \tau \frac{dN(t)}{dt}.$$

Thus, we have

$$\begin{aligned} \frac{dN}{dt} &= N(\mu_N - \gamma_1 L - \gamma_2 X), \\ \frac{dL}{dt} &= (L^* - L)(\mu_L - \gamma_3 N(1 + \tau(\mu_N - \gamma_1 L - \gamma_2 X)) - \gamma_4 G). \end{aligned} \quad (3)$$

Stationary points found from the system:

Колянова Т.В.¹, асистент

Вплив запізнення на характер стійкості стаціонарних точок в математичній моделі надбаного імунітету

Розглядається математична модель надбаного імунітету, яка описується системою нелінійних диференціальних рівнянь із запізненням. Аналізується вплив запізнення на стійкість стаціонарних точок, наводяться біологічні наслідки.

Ключові слова: система нелінійних диференціальних рівнянь, імунітет, антиген, стійкість стаціонарного стану.

¹ Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Глушкова 4д,

e-mail: tania.kolianova@gmail.com

$$\begin{aligned} N(\mu_N - \gamma_1 L - \gamma_2 X) &= 0, \\ (L^* - L)(\mu_L - \gamma_3 N(1 + \tau(\mu_N - \gamma_1 L - \gamma_2 X)) - \gamma_4 G) &= 0. \end{aligned} \quad (4)$$

The first stationary point

$$N_1 = 0, L_1 = L^*, \quad (5)$$

describes a healthy organism, and the second stationary point

$$L_2 = \frac{\mu_N - \gamma_2 X}{\gamma_1}, N_2 = \frac{\mu_L - \gamma_4 G}{\gamma_3}, \quad (6)$$

responds for a sick organism.

Investigate the stability of stationary points.

Theorem 1. A stationary point (5) will be asymptotic stable at implementation of such condition $\mu_N - \gamma_2 X < \gamma_1 L^*$, if the condition $\mu_N - \gamma_2 X > \gamma_1 L^*$, the stationary point is unstable, and if $\mu_N - \gamma_2 X = \gamma_1 L^*$, the stationary point is stable.

Proof:

Linearized system of differential equations (3) in around of this point (5) will look like

$$\begin{aligned} \frac{dN}{dt} &= (\mu_N - \gamma_1 L^* - \gamma_2 X)N, \\ \frac{d\tilde{L}}{dt} &= (\gamma_4 G - \mu_L)\tilde{L}, \end{aligned} \quad (7)$$

where $\tilde{L} = L - L_1$.

We write the matrix coefficients for the linearized system

$$A_1 = \begin{pmatrix} \mu_N - \gamma_1 L^* - \gamma_2 X & 0 \\ 0 & \gamma_4 G - \mu_L \end{pmatrix}.$$

Characteristic equation $\det(A_1 - \lambda E) = 0$ will look like

$$\begin{vmatrix} (\mu_N - \gamma_1 L^* - \gamma_2 X) - \lambda & 0 \\ 0 & (\gamma_4 G - \mu_L) - \lambda \end{vmatrix} = 0$$

or

$$((\mu_N - \gamma_1 L^* - \gamma_2 X) - \lambda)((\gamma_4 G - \mu_L) - \lambda) = 0,$$

here received

$$\begin{aligned} \lambda_1 &= \mu_N - \gamma_1 L^* - \gamma_2 X, \\ \lambda_2 &= \gamma_4 G - \mu_L. \end{aligned}$$

Clearly, that $\lambda_2 < 0$ always (due to the fact that $N_2 > 0$). As for the sign λ_1 , then if $\lambda_1 < 0$ – singular point is asymptotic stable, if $\lambda_1 > 0$ – singular point is unstable, and if $\lambda_1 = 0$ – singular point is stable.

The theorem is proved.

Results. In the case of the stable node ($\lambda_1 < 0$, $\lambda_2 < 0$), the existing population of lymphocytes in the body exceeds the speed of propagation of antigen and with it the content of corticosteroids G did not significantly affect the rate of reproduction and recovery of lymphocytes in the body [1]. Under this situation, the immune system described situation is stable and even a small perturbation of the steady state does not put it out of balance. Human remains healthy.

In the case of the saddle ($\lambda_1 > 0$, $\lambda_2 < 0$), despite the fact that the content of corticosteroids does not substantially affect the reproduction rate and the recovery of lymphocytes in the body, yet the rate of reproduction and dissemination of antigen exceeds the lymphocyte population in the body (weak immune status), which can result in chronic disease.

Theorem 2. For $\tau > 0$, if the condition is $\gamma_1 L^* = \mu_N - \gamma_2 X$, then a stationary point (6) is a type of unstable degenerate node, if

$\gamma_1 L^* < \mu_N - \gamma_2 X$, then stationary point – is unstable focus, if $\gamma_1 L^* > \mu_N - \gamma_2 X$, then the stationary point (6) is saddle.

Proof:

Linearized system of differential equations (3) in around of the second point (6) will look like

$$\begin{aligned} \frac{d\tilde{N}}{dt} &= \frac{\gamma_1}{\gamma_3}(\mu_L - \gamma_4 G)\tilde{L}, \\ \frac{d\tilde{L}}{dt} &= -\gamma_3(L^* - \frac{\mu_N - \gamma_2 X}{\gamma_1})\tilde{N} + \tau(\mu_L - \gamma_4 G)(\mu_N - \gamma_2 X)\tilde{L}, \end{aligned}$$

where $\tilde{L} = L - L_2$, $\tilde{N} = N - N_2$.

Matrix coefficients for the linearized system has the form

$$A_2 = \begin{pmatrix} 0 & -\frac{\gamma_1}{\gamma_3}(\mu_L - \gamma_4 G) \\ -\gamma_3(L^* - \frac{\mu_N - \gamma_2 X}{\gamma_1}) & \tau(\mu_L - \gamma_4 G)(\mu_N - \gamma_2 X) \end{pmatrix}$$

Characteristic equation $\det(A_2 - \lambda E) = 0$ will look like

$$\begin{vmatrix} -\lambda & -\frac{\gamma_1}{\gamma_3}(\mu_L - \gamma_4 G) \\ -\gamma_3(L^* - \frac{\mu_N - \gamma_2 X}{\gamma_1}) & \tau(\mu_L - \gamma_4 G)(\mu_N - \gamma_2 X) - \lambda \end{vmatrix}.$$

Then we have

$$\begin{aligned} \lambda^2 - \lambda\tau(\mu_L - \gamma_4 G)(\mu_N - \gamma_2 X) - \\ - \gamma_1(\mu_L - \gamma_4 G)(L^* - \frac{\mu_N - \gamma_2 X}{\gamma_1}) = 0. \end{aligned}$$

Roots of this quadratic equation have the form

$$\begin{aligned} \lambda_1 &= \frac{\tau(\mu_L - \gamma_4 G)(\mu_N - \gamma_2 X)}{2} + \\ &+ \sqrt{\gamma_1(\mu_L - \gamma_4 G)(L^* - \frac{\mu_N - \gamma_2 X}{\gamma_1})}, \\ \lambda_2 &= \frac{\tau(\mu_L - \gamma_4 G)(\mu_N - \gamma_2 X)}{2} - \\ &- \sqrt{\gamma_1(\mu_L - \gamma_4 G)(L^* - \frac{\mu_N - \gamma_2 X}{\gamma_1})}. \end{aligned} \quad (8)$$

We must evaluate the sign

$$D_1 = \gamma_1(\mu_L - \gamma_4 G)(L^* - \frac{\mu_N - \gamma_2 X}{\gamma_1}).$$

Because of $\tau > 0$, then we have for $D_1 = 0$:

$$\lambda_1 = \lambda_2 = \frac{\tau(\mu_L - \gamma_4 G)(\mu_N - \gamma_2 X)}{2} > 0 \quad - \quad \text{the}$$

roots are multiples and positive, so we have an unstable degenerate node;

for $D_1 < 0$

roots of the characteristic equation are

$$\lambda_{1,2} = \frac{\tau(\mu_L - \gamma_4 G)(\mu_N - \gamma_2 X)}{2} \pm i \sqrt{\gamma_1(\mu_L - \gamma_4 G)(L^* - \frac{(\mu_N - \gamma_2 X)}{\gamma_1})}$$

and $\text{Re} \lambda > 0$, therefore, we have an unstable focus;

for $D_1 > 0$, then $\lambda_1 > 0$, $\lambda_2 < 0$ – stationary point is saddle, but if τ grows, we have $(\lambda_1 > 0, \lambda_2 > 0)$ an unstable node.

The theorem is proved.

Results. In the case of a singular point of saddle - the body or be able to overcome the disease, or may be lost. In the case where the singular point is the center, the body will have a chronic disease that periodically will escalate.

In the case of a singular point of the body of an unstable focus will continue to experience some “hesitation”: the aggravation of the disease and a

weakened immune system (including the reduction of hormones (corticosteroids)) that suppress the production of lymphocytes), and then will increase the production of lymphocytes and a slight weakening of the disease. With increasing content of lymphocytes increased hormone (G), which will result inhibit the production of lymphocytes and lead to a new outbreak. Such constant stress on the immune system result in an exhaustion [1, 2], whereby the body can not cope with the disease and die.

In the case where the singular point is an unstable node, depending on the state of the immune system (the system parameters and the ratio of (3)), or the body can overcome the disease or inhibit the growth of populations of antigen specific “long” period of time, or in the case of a weakened immune system - will be the rapid growth of the population of antigens, the body with the disease eventually can not cope and die [3, 4].

Список використаних джерел

1. Bernet F. Cellular immunology. M., «World», 1971. (in Russian).
2. Hege J. S. A mathematical model relating circulating antibody and antibody forming cells / J.S. Hege, G. Cole // J. Immunol. 1966. V. 97. P. 34-40. (in English).
3. Khayrer E., Nerset P., Vanner G. Decision of usual differential equalizations. Non-rigid tasks. Moscow, 1990. (in Russian).
4. Pontryagin L.S. Usual differential equalizations. L.S.Pontryagin. M.: Science, 1974. - 332 p. (in Russian).

Надійшла до редколегії 14.05.13

УДК 519.713+530.145

Скобелев В.Г., д.ф.-м.н., д.т.н., професор

Квантові автомати з операторами які комутують

У статті досліджено множини квантових скінченних автоматів у припущенні, що унітарні оператори, які асоційовано з вхідним алфавітом автомату, комутують один з одним. У термінах вхідної фактор-напівгрупи, яку визначено на основі вказаної властивості унітарних операторів охарактеризовано мови, які моделі дослідження розпізнають або з заданою ймовірністю, або з заданою похибкою. Ключові слова: квантові скінченні автомати, комутуючі унітарні оператори.

Інститут прикладної математики і механіки
НАН України, 83114, м. Донецьк, вул. Розі
Люксембург, 74,
e-mail: skbv@iamm.ac.donetsk.ua

V. G. Skobelev, Dr. (Phys.-Math.) Sci, Prof.

Quantum automata with operators that commutes

The sets of quantum finite automata under assumption that unitary operators associated with input alphabet of an automaton commutes each with the others are investigated in the given paper. Languages accepted by investigated models either with given probability, or with given mistake are characterized in terms of the factor-semigroup determined by the above pointed property of unitary operators. Key Words: quantum finite automata, unitary operators that commute each with the others.

Institute of Applied Mathematics and Mechanics of
NAS of Ukraine, 83114, Rose Luxemburg str., 74,
e-mail: skbv@iamm.ac.donetsk.ua

Статтю представив чл.-кор. НАН України, проф. Анісімов А.В.

Introduction.

Development of quantum algorithms theory has stimulated research of quantum automata (QA). There were proposed a variety of different models of QA intended to recognize languages in the given alphabet (i.e. models that are acceptors). These models differ essentially in their complexity and recognizing capacities. Main efforts of researchers were directed on attempts to determine the class of languages recognized by this or the other model in explicit form, to compare recognizing capacities of proposed models each with the others as well as with classic deterministic or probability automata, and to establish criteria for equivalent states of QA.

The main aim of the given paper is to investigate existing models of finite QA (in what follows the word "finite" is omitted, as a rule) under assumption that unitary operators associated with input alphabet of an automaton commutes each with the others.

The rest of the paper is organized as follows. In chapter 1 basic notions connected with quantum systems are presented. In chapter 2 existing models of QA are considered briefly. In chapter 3 the sets of finite QA under assumption that unitary operators associated with input alphabet of an automaton commutes each with the others are investigated. In terms of input factor-semigroup determined by the

above pointed property of unitary operators there are characterized languages accepted by investigated models of finite QA either with given probability, or with given mistake. The last chapter consists of some conclusion remarks.

1. Basic notions.

Let $\mathbf{C}^n$ ($n \in \mathbf{N}, n \geq 2$) be n -dimensional complex vector space of elements

$$|\varphi\rangle = (\alpha_1, \dots, \alpha_n)^T$$

with computational basis

$$\mathbf{B}_n = \{|i\rangle | i \in \mathbf{N}_n\},$$

where

$$|i\rangle = (\underbrace{0, \dots, 0}_{i-1}, 1, \underbrace{0, \dots, 0}_{n-i})^T.$$

Scalar product is defined as

$$\langle \varphi | \psi \rangle = \langle \varphi | \cdot | \psi \rangle,$$

where $\langle \varphi | = | \varphi \rangle^\dagger$, and $\dagger$ is operation of Hermitian conjugation. The norm of any element $|\varphi\rangle \in \mathbf{C}^n$ is defined as

$$\| |\varphi\rangle \| = \sqrt{\langle \varphi | \varphi \rangle}.$$

In what follows any linear operator in $\mathbf{C}^n$ is presented via its matrix in computational basis.

In quantum computing we are restricted with unit vectors $|\varphi\rangle \in \mathbb{C}^n$ (i.e. $\| |\varphi\rangle \| = 1$), unitary operators U in $\mathbb{C}^n$ (i.e. $U^\dagger U = I$) and measurements.

In what follows we deal with complete projective measurements only, i.e. with sets

$$P = \{P_i\}_{i \in \mathbf{N}_m}$$

of pair-wise orthogonal operators in $\mathbb{C}^n$, such that

$$\sum_{i=1}^m P_i = I.$$

In the result of measurement P any unit vector $|\varphi\rangle \in \mathbb{C}^n$ is transformed into unit vector

$$|\psi_i\rangle = \frac{P_i |\varphi\rangle}{\sqrt{\langle \varphi | P_i^\dagger P_i | \varphi \rangle}}$$

with probability

$$p_i = \langle \varphi | P_i^\dagger P_i | \varphi \rangle.$$

It is worth to note that unit vectors $|\psi_i\rangle$ ($i \in \mathbf{N}_m$) are pair-wise orthogonal and $\sum_{i=1}^m p_i = 1$ (completeness equation).

Remark 1. Any partition

$$\pi = \{C_1, \dots, C_h\} \quad (1 \leq h \leq n)$$

of computational basis B_n defines projective measurement $P = \{P_i\}_{i \in \mathbf{N}_h}$, where P_i ($i \in \mathbf{N}_h$) is projection operator on the subspace spanned by C_i , i.e.

$$P_i = \sum_{|j\rangle \in C_i} |j\rangle \langle j|.$$

It is worth to note that $P_i^\dagger = P_i$ and $P_i^2 = P_i$.

Computing produced by quantum system S in $\mathbb{C}^n$ is defined as any finite sequence of unitary transformations and measurements applied either to some initial pure state (i.e. to some unit vector $|\varphi\rangle \in \mathbb{C}^n$), or to some initial mixed state, i.e. to some set

$$S = \{(|\psi_i\rangle, p_i)\}_{i \in \mathbf{N}_l} \quad (p_i > 0, \sum_{i \in \mathbf{N}_l} p_i = 1),$$

where $|\psi_i\rangle \in \mathbb{C}^n$ ($i \in \mathbf{N}_l$) are pair-wise different unit vectors and p_i ($i \in \mathbf{N}_l$) is probability that at initial instant the system S exists in the state $|\psi_i\rangle$.

Remark 2. Any mixed state $\{(|\psi_i\rangle, p_i)\}_{i \in \mathbf{N}_l}$ of quantum system S can be presented via density operator

$$\rho = \sum_{i=1}^l p_i |\psi_i\rangle \langle \psi_i|.$$

Thus, any unitary operator U transforms mixed state ρ into the mixed state $U^\dagger \rho U$.

The result of computing produced by quantum system S is the result of the last measurement.

Remark 3. Thus, any algorithm realized by any quantum system S is probabilistic one in its essence, since any unit vector $|\varphi\rangle \in \mathbb{C}^n$ in the result of any measurement is transformed into unit vector $|\psi_i\rangle$ with probability p_i .

2. Models of QA.

We consider existing models of QA briefly.

Remark 4. All models of QA are defined as acceptors, and are presented in terms of quantum Turing machine (QTM).

Let X ($|X| = m$) be fixed input alphabet. We associate with each letter $x \in X$ some unitary operator U_x in $\mathbb{C}^n$.

The model MO-1QFA [1] is the simplest model of QA. It is defined to be 1-way 1-head QTM

$$M = (Q, X, |\varphi_0\rangle, Q_{acc}),$$

where $Q = B_n$ is the set of basic states, the unit vector $|\varphi_0\rangle \in \mathbb{C}^n$ is pure initial state and $Q_{acc} \subseteq Q$ is the set of accepting states.

Let a string $\$x_1 \dots x_l \#$ (where $x_1 \dots x_l \in X^*$, while $\$$ and $\#$ is, correspondingly, left and right marker) be written in the input tape of QTM M . At initial instant QTM M exists in the state $|\varphi_0\rangle$ and the head of QTM M observes the cell in which left marker $\$$ is written.

Computing produced by QTM M consists of the following $l+2$ steps:

1. At the 1st instant the head of QTM M moves by one cell to the right (thus, after $i-1$ instants ($i=2, \dots, l+1$) QTM M exists in the state $|\varphi_{i-1}\rangle$ and its head observes the cell in which x_i is written).

2. At instant i ($i=2, \dots, l+1$) QTM M acts as follows. If $x_i = \Lambda$ (where Λ is the empty symbol) then the head of QTM M moves by one cell to the right (thus, $|\varphi_i\rangle = |\varphi_{i-1}\rangle$). If $x_i \in X$ then the state $|\varphi_{i-1}\rangle$ of QTM M is transformed in the state $|\varphi_i\rangle = U_{x_i} |\varphi_{i-1}\rangle$ and the head of QTM M moves by one cell to the right.

3. At instant $l+2$ (after $l+1$ instants QTM M exists in the state $|\varphi_l\rangle$ and its head observes the cell in which $\#$ is written) the state $|\varphi_l\rangle$ is measured in the basis B_n and QTM M halts.

If in the result of this measurement we get some element of the set Q_{acc} then the string $x_1 \dots x_l \in X^*$ is accepted, otherwise it is rejected.

Probability that QTM M with pure initial state $|\varphi_0\rangle$ accepts any string $w = x_1 \dots x_l$ equals to

$$P_M(|\varphi_0\rangle, w) = \|P_{acc} U_w |\varphi_0\rangle\|^2,$$

where $U_w = U_{x_l} \dots U_{x_1}$ and P_{acc} is the projection operator on the subspace spanned by Q_{acc} , i.e.

$$P_{acc} = \sum_{|i\rangle \in Q_{acc}} |i\rangle \langle i|.$$

Thus, MO-1QFA can be interpreted either as acceptor of a language with given probability, or as acceptor of a language with given mistake.

Remark 5. The language L is accepted:

1) with probability p ($0.5 < p \leq 1$), if any string $w \in L$ is accepted with probability not less than p , while any string $w \notin L$ is accepted with probability not exceeding $1 - p$;

2) with given mistake (p_1, p_2) ($0 \leq p_1 < p_2 < 1$), if any string $w \in L$ is accepted with probability not less than p_2 , while any string $w \notin L$ is accepted with probability not exceeding p_1 .

It is established in [2] that the set of languages accepted by the model MO-1QFA with given mistake equals to the set L_{GFA} of languages accepted by group finite automata.

The model MM-1QFA [3] differs from the model MO-1QFA in the following way:

1) there are extracted non-intersecting subsets Q_{rej} and Q_{nt} , correspondingly, of rejecting and of non-accepting basic states ($Q_{rej} \cup Q_{nt} = Q \setminus Q_{acc}$);

2) at any intermediate instant it is admissible application of corresponding unitary operator followed by measurement in the basis B_n ;

3) at any intermediate instant if the result of measurement is some basic state $q \in Q_{nt}$ then the next instant starts, while if the result of measurement is some basic state $q \in Q_{acc} \cup Q_{rej}$ then QTM M halts (analyzed string is accepted, if $q \in Q_{acc}$ and rejected, if $q \in Q_{rej}$).

Remark 6. The model MM-1QFA is based on 'Decide and Halt' approach intended to solve some class of 'problems of recognition' called 'promise problems'.

These problems are characterized as follows. It is defined some language $L \subseteq \{0,1\}^*$. Analyzed 'problem of recognition' is reduced to the problem of

accepting any input to which corresponds some string $w \in L$ and of rejecting any input to which corresponds some string $w \notin L$. Two non-intersecting languages $L_{YES} \subseteq \{0,1\}^*$ and $L_{NO} \subseteq \{0,1\}^*$ are also defined. Union of these languages is called 'promise' and is intended to halt algorithm at intermediate instants. If at intermediate instant algorithm processes some fragment of input to which some string $v \in L_{YES} \cup L_{NO}$ corresponds then it halts (input is accepted, if $v \in L_{YES}$ and is rejected, if $v \in L_{NO}$). Otherwise (i.e., if $v \notin L_{YES} \cup L_{NO}$), algorithm at next instant forms next fragment of input and processed it as described above.

Till now, the set $S_{MM-1QFA}$ of languages accepted by the model MM-1QFA is not described in explicit form. It is known only that there hold inclusions $L_{GFA} \subset S_{MM-1QFA} \subset L_{DFA}$ (where L_{DFA} is the set of languages accepted by deterministic finite automata). Also it is established in [4] that the set $S_{MM-1QFA}$ is not closed under Boolean operations.

The model L-QFA [5] deals with mixed states. The set S_{L-QFA} of languages accepted by the model L-QFA equals to the set L_{IPFA} of languages accepted by invertible probabilistic finite automata. This set is closed under Boolean operations.

The model N-QFA [6] at any instant admits application of any sequence of unitary operators and projective measurements. Thus, exploring of ancilla qubits can be easily presented in this model. Till now, the set S_{N-QFA} of languages accepted by the model N-QFA is not described in explicit form. It is known only that $L_{GFA} \subset S_{N-QFA} \subset L_{DFA}$.

The model k QFA ($k \in \mathbb{N}$) [7] is defined to be any 1-way k -head QTM

$$M = (Q, T, \mu, |\varphi_0\rangle, Q_{acc}),$$

where

$$T = \{\$ \} X^{k-1} \cup \{\Lambda\}^k \cup \bigcup_{i=1}^{k-1} \{\Lambda\}^{k-i} X^i \cup X^k \cup X^{k-1} \{\#\}$$

and μ is some mapping of the set T into the set of unitary operators in $\mathbb{C}^n$.

It is supposed that at any instant all heads of QTM M move simultaneously by one cell to the right. Till now, the set S_{QFA_k} of languages accepted by the model k QFA is not described in explicit form. It is established that the set S_{QFA_k} is closed under Boolean operations, but is not closed under

concatenation and iteration. Also it is established, and that there hold inclusions $S_{QFA_k} \subset S_{QFA_{k+1}}$ and

$$S_{QFA^*} \subset S_{DFA}, \text{ where } S_{QFA^*} = \bigcup_{k=1}^{\infty} S_{QFA_k}.$$

The model CL-QFA [8] is generalization of the model N-QFA. In this model some regular language in the alphabet of eigenvalues of Hermitian operators is used to determine projective measurements at intermediate instants.

Till now, the set S_{CL-QFA} of languages accepted by the model CL-QFA is not described in explicit form. It is known only that this model can accept (with given mistake) any language $L \in L_{DFA}$.

The model 2-QFA [3] is defined in terms of any 2-way 1-head QTM under supposition that at any instant this model determines direction in which the head of QTM is shifted to the neighboring cell.

The model 1,5-QFA [9] is defined in terms of any 1-way 1-head QTM under supposition that at any instant this model determines either the head of QTM is motionless, or it is shifted to the right neighboring cell.

Till now, the set S_{2-QFA} of languages accepted by the model 2-QFA, as well as the set $S_{1,5-QFA}$ of languages accepted by the model 1,5-QFA is not described in explicit form. It is known only that there hold inclusions $L_{DFA} \subset S_{2-QFA}$ and $L_{DFA} \subset S_{1,5-QFA}$. It is also established in [10] that it is algorithmically unsolvable the problem of checking if it is empty a language accepted by 1,5-QFA and 2-QFA, both.

The model C-QFA [11] is generalization of the model 1,5-QFA under supposition that at any instant this model can also write this or the other input symbol into observed cell. This model was not investigated in details.

Different models of 1-QFA, 1,5-QFA and 2-QFA with counters were defined in [12-15].

Till now, the sets of languages accepted by these models are not described in explicit form. It is known only that the set of languages accepted by probabilistic finite automata is proper subset of the sets of languages accepted by these models.

It is worth to note that any of models CL-QFA, 1,5-QFA, 2-QFA, C-QFA and QFA with counters explores some technique that is not used for classic finite automata. Thus, it is not quite correct to consider these models as quantum finite automata.

3. Main results.

In this chapter it is supposed that unitary operators associated with investigated model of

quantum finite automaton commute each with the others.

Firstly, we characterize languages accepted by the model MO-1QFA, and by the model L-QFA with measurement at final instant only.

Let $X = \{x_1, \dots, x_m\}$ be the input alphabet of QA. Unitary operator associated with input symbol $x_i \in X$ is denoted by U_i . In what follows it is supposed that

$$U = \{U_i \mid i \in \mathbf{N}_m\}$$

is any set of unitary operators that commute with each other, i.e.

$$U_i U_j = U_j U_i \quad (i, j \in \mathbf{N}_n).$$

We associate with any input string $w \in X^l$ ($l \in \mathbf{N}$) the string

$$pr_U(w) = U_1^{r_1} \dots U_m^{r_m} \quad (1)$$

of unitary operators, where r_i ($i \in \mathbf{N}_m$) is the number of occurrences of the letter x_i in the string w (thus, $r_i \geq 0$ ($i \in \mathbf{N}_m$) and $\sum_{i=1}^m r_i = l$).

Let $\equiv_{X,U}$ be equivalence on X^+ defined by identity

$$w_1 \equiv_{X,U} w_2 \Leftrightarrow pr_U(w_1) = pr_U(w_2). \quad (2)$$

Theorem 1. Let U be any set of unitary operators that commute with each other. Then any language accepted (either with given probability, or with given mistake) by the model MO-1QFA, as well as by the model L-QFA with measurement at final instant only is union of some elements of the factor-set $X^+ / \equiv_{X,U}$.

Proof. We set

$$prod(pr_U(w)) = U_1^{r_1} \dots U_m^{r_m}. \quad (3)$$

Since U is a set of unitary operators that commute with each other then formulae (1) and (3) imply that

$$prod(pr_U(w)) = U_w \quad (4)$$

for all input strings $w \in X^+$.

Formulae (2) and (4) imply that with any element $B \in X^+ / \equiv_{X,U}$ it can be associated unitary operator U_B , such that for any input string $w \in B$:

1) for any initial pure state $|\varphi_0\rangle \in \mathbf{C}^n$ of the model MO-1QFA it holds identity

$$U_B |\varphi_0\rangle = U_w |\varphi_0\rangle; \quad (5)$$

2) for any density operator ρ of the model L-QFA with measurement at final instant only it holds identity

$$U_B^\dagger \rho U_B = U_w^\dagger \rho U_w. \quad (6)$$

Remark 7. Identity (5) and (6) establish that transformations of any initial state of the models MO-1QFA or L-QFA under actions of input strings are completely determined by the set of unitary operators U_B ($B \in X^+ / \equiv_{X,U}$). This factor implies that any proposition for the models MO-1QFA and L-QFA formulated only in terms of transformations of initial state under actions of input strings can be reformulated in equivalent proposition formulated in terms of unitary operators U_B ($B \in X^+ / \equiv_{X,U}$).

Identity (5) implies that for any initial pure state $|\varphi_0\rangle \in \mathbf{C}^n$ of the model MO-1QFA and for any input string $w \in B$ it holds identity

$$\begin{aligned} P_M(|\varphi_0\rangle, w) &= \|P_{acc}U_w|\varphi_0\rangle\|^2 = \\ &= \|P_{acc}U_B|\varphi_0\rangle\|^2. \end{aligned} \quad (7)$$

Identity (7) implies that for any initial pure state $|\varphi_0\rangle \in \mathbf{C}^n$ of the model MO-1QFA and for any element $B \in X^+ / \equiv_{X,U}$ it is correct to define

$$P_M(|\varphi_0\rangle, B) = P_M(|\varphi_0\rangle, w) \quad (w \in B). \quad (8)$$

For any initial pure state $|\varphi_0\rangle \in \mathbf{C}^n$ of the model MO-1QFA and any number p ($0.5 < p \leq 1$) we set

$$\begin{aligned} B_M^{(1)}(|\varphi_0\rangle, p) &= \\ &= \{B \in X^+ / \equiv_{X,U} | P_M(|\varphi_0\rangle, B) \leq 1 - p\}, \quad (9) \\ B_M^{(2)}(|\varphi_0\rangle, p) &= \\ &= \{B \in X^+ / \equiv_{X,U} | P_M(|\varphi_0\rangle, B) \geq p\}. \end{aligned} \quad (10)$$

Identities (8)-(10) imply that the language accepted with given probability p ($0.5 < p \leq 1$) by the model MO-1QFA existing in initial pure state $|\varphi_0\rangle \in \mathbf{C}^n$ is non-empty if and only if it holds identity

$$B_M^{(1)}(|\varphi_0\rangle, p) \cup B_M^{(2)}(|\varphi_0\rangle, p) = X^+ / \equiv_{X,U}.$$

Moreover, this language is defined by identity

$$L_M(|\varphi_0\rangle, p) = \bigcup_{B \in B_M^{(2)}(|\varphi_0\rangle, p)} B. \quad (11)$$

In a similar way for any initial pure state $|\varphi_0\rangle \in \mathbf{C}^n$ of the model MO-1QFA and any ordered pair of numbers (p_1, p_2) ($0 \leq p_1 < p_2 < 1$) we set

$$\begin{aligned} \tilde{B}_M^{(1)}(|\varphi_0\rangle, p_2) &= \\ &= \{B \in X^+ / \equiv_{X,U} | P_M(|\varphi_0\rangle, B) \leq p_1\}, \quad (12) \\ \tilde{B}_M^{(2)}(|\varphi_0\rangle, p_1) &= \\ &= \{B \in X^+ / \equiv_{X,U} | P_M(|\varphi_0\rangle, B) \geq p_2\}. \end{aligned} \quad (13)$$

Identities (8), (12) and (13) imply that the language accepted with given mistake (p_1, p_2) ($0 \leq p_1 < p_2 < 1$) by the model MO-1QFA existing in

initial pure state $|\varphi_0\rangle \in \mathbf{C}^n$ is non-empty if and only if it holds identity

$$\tilde{B}_M^{(1)}(|\varphi_0\rangle, p_1) \cup \tilde{B}_M^{(2)}(|\varphi_0\rangle, p_2) = X^+ / \equiv_{X,U}.$$

Moreover, this language is defined by identity

$$L_M(|\varphi_0\rangle, (p_1, p_2)) = \bigcup_{B \in \tilde{B}_M^{(2)}(|\varphi_0\rangle, p_2)} B. \quad (14)$$

Now we consider in a similar way the model L-QFA existing in initial mixed state $\{(|\psi_i\rangle, p_i)\}_{i \in N_I}$ ($p_i > 0, \sum_{i \in N_I} p_i = 1$). Probability that in the result of

transformation associated with input sequence $w \in X^+$ and measurement in the computational basis B_n at final instant the model L-QFA would exist in some basic state $|i\rangle \in Q_{acc}$ is denoted by $P_M(\{(|\psi_i\rangle, p_i)\}_{i \in N_I}, w)$.

Since $|\psi_i\rangle \in \mathbf{C}^n$ ($i \in N_I$) are pair-wise different unit vectors we get that for any input string $w \in X^+$

$$\begin{aligned} P_M(\{(|\psi_i\rangle, p_i)\}_{i \in N_I}, w) &= \sum_{i \in N_I} p_i \|P_{acc}U_w|\psi_i\rangle\|^2 = \\ &= \sum_{i \in N_I} p_i P_M(|\psi_i\rangle, w). \end{aligned} \quad (15)$$

Identities (8) and (15) imply that it is correct to define

$$\begin{aligned} P_M(\{(|\psi_i\rangle, p_i)\}_{i \in N_I}, B) &= \\ &= P_M(\{(|\psi_i\rangle, p_i)\}_{i \in N_I}, w) \quad (w \in B). \end{aligned} \quad (16)$$

For any initial mixed state $\{(|\psi_i\rangle, p_i)\}_{i \in N_I}$ ($p_i > 0, \sum_{i \in N_I} p_i = 1$) of the model L-QFA and any number p ($0.5 < p \leq 1$) we set

$$\begin{aligned} \tilde{B}_M^{(1)}(\{(|\psi_i\rangle, p_i)\}_{i \in N_I}, p) &= \\ &= \{B \in X^+ / \equiv_{X,U} | P_M(\{(|\psi_i\rangle, p_i)\}_{i \in N_I}, B) \leq 1 - p\}, \\ \tilde{B}_M^{(2)}(\{(|\psi_i\rangle, p_i)\}_{i \in N_I}, p) &= \\ &= \{B \in X^+ / \equiv_{X,U} | P_M(\{(|\psi_i\rangle, p_i)\}_{i \in N_I}, B) \geq p\}. \end{aligned}$$

Identity (16) imply that the language accepted with given probability p ($0.5 < p \leq 1$) by the model L-QFA existing in initial mixed state $\{(|\psi_i\rangle, p_i)\}_{i \in N_I}$ ($p_i > 0, \sum_{i \in N_I} p_i = 1$) is non-empty if and only if it holds identity

$$\begin{aligned} \tilde{B}_M^{(1)}(\{(|\psi_i\rangle, p_i)\}_{i \in N_I}, p) &\cup \\ \cup \tilde{B}_M^{(2)}(\{(|\psi_i\rangle, p_i)\}_{i \in N_I}, p) &= X / \equiv_{X,U}. \end{aligned}$$

Moreover, this language is defined by identity

$$L_M(\{(|\psi_i\rangle, p_i)\}_{i \in N_I}, p) = \bigcup_{B \in \tilde{B}_M^{(2)}(\{(|\psi_i\rangle, p_i)\}_{i \in N_I}, p)} B. \quad (17)$$

For any initial mixed state $\{(|\psi_i\rangle, p_i)\}_{i \in \mathbf{N}_I}$ ($p_i > 0, \sum_{i \in \mathbf{N}_I} p_i = 1$) of the model L-QFA and any ordered pair of numbers $(\tilde{p}_1, \tilde{p}_2)$ ($0 \leq \tilde{p}_1 < \tilde{p}_2 < 1$) we set

$$\begin{aligned} \hat{B}_M^{(1)}(\{(|\psi_i\rangle, p_i)\}_{i \in \mathbf{N}_I}, \tilde{p}_2) &= \\ = \{B \in X^+ / \equiv_{X,U} | P_M(\{(|\psi_i\rangle, p_i)\}_{i \in \mathbf{N}_I}, B) \leq \tilde{p}_1\}, \\ \hat{B}_M^{(2)}(\{(|\psi_i\rangle, p_i)\}_{i \in \mathbf{N}_I}, \tilde{p}_1) &= \\ = \{B \in X^+ / \equiv_{X,U} | P_M(\{(|\psi_i\rangle, p_i)\}_{i \in \mathbf{N}_I}, B) \leq \tilde{p}_1\}. \end{aligned}$$

Thus, the language accepted with given mistake $(\tilde{p}_1, \tilde{p}_2)$ ($0 \leq \tilde{p}_1 < \tilde{p}_2 < 1$) by the model L-QFA existing in initial mixed state $\{(|\psi_i\rangle, p_i)\}_{i \in \mathbf{N}_I}$ ($p_i > 0, \sum_{i \in \mathbf{N}_I} p_i = 1$) is non-empty if and only if it holds identity

$$\begin{aligned} \hat{B}_M^{(1)}(\{(|\psi_i\rangle, p_i)\}_{i \in \mathbf{N}_I}, \tilde{p}_1) \cup \\ \cup \hat{B}_M^{(2)}(\{(|\psi_i\rangle, p_i)\}_{i \in \mathbf{N}_I}, \tilde{p}_2) = X^+ / \equiv_{X,U}. \end{aligned}$$

Moreover, this language is defined by identity

$$\begin{aligned} L_M(\{(|\psi_i\rangle, p_i)\}_{i \in \mathbf{N}_I}, (\tilde{p}_1, \tilde{p}_2)) = \\ = \bigcup_{B \in \hat{B}_M^{(2)}(\{(|\psi_i\rangle, p_i)\}_{i \in \mathbf{N}_I}, \tilde{p}_2)} B. \end{aligned} \quad (18)$$

Identities (11), (14), (17) and (18) imply that theorem holds.

Q.E.D.

Corollary 1. Let for the model k QFA with measurement at final instant only the set $U = \text{Val } \mu$ be any set of unitary operators that commute with each other. Then any language accepted (either with given probability, or with given mistake) by the model k QFA is union of some elements of the factor-set $T^+ / \equiv_{T,U}$.

Proof. To prove corollary 1 it is sufficient to substitute T instead of X in proof of theorem 1.

Q.E.D.

Let $\equiv'_{X,U}$ be any equivalence on the set X , such that every element of the factor-set $X^+ / \equiv'_{X,U}$ is union of some elements $B \in X^+ / \equiv_{X,U}$ to which the same unitary operator U_B is associated.

Corollary 2. Let U be any set of unitary operators that commute with each other. Then any language accepted (either with given probability, or with given mistake) by the model MO-1QFA, as well as by the model L-QFA with measurement at final instant only is union of some elements of the factor-set $X^+ / \equiv'_{X,U}$.

Proof. Identities (4)-(6) justify that without any restrictions equivalence $\equiv'_{X,U}$ can be substituted instead of equivalence $\equiv_{X,U}$ in proof of theorem 1.

Q.E.D.

Important special case of equivalence $\equiv'_{X,U}$ on the set X takes the place in the following situation.

Let $U = \{U_i | i \in \mathbf{N}_m\}$ be a set of unitary operators that commute with each other, such that for every $i \in \mathbf{N}_m$ there exists some integer a_i that satisfies identity

$$U_i^{a_i} = I. \quad (19)$$

Let a_i be minimal positive integer that satisfies to identity (19). We define equivalence $\equiv_{X,U}^{(1)}$ on the set X^+ in the following way: for any input strings $w_1, w_2 \in X^+$ ($pr_U(w_i) = U_1^{r_{i1}} \dots U_m^{r_{im}}$ ($i = 1, 2$))

$$\begin{aligned} w_1 \equiv_{X,U}^{(1)} w_2 \Leftrightarrow U_1^{r_{11}(\text{mod } a_1)} \dots U_m^{r_{m1}(\text{mod } a_m)} = \\ = U_1^{r_{12}(\text{mod } a_1)} \dots U_m^{r_{m2}(\text{mod } a_m)}. \end{aligned} \quad (20)$$

Identities (19) and (20) imply that equivalence $\equiv_{X,U}^{(1)}$ is special case of equivalence $\equiv'_{X,U}$. Moreover, it holds identity

$$|X / \equiv_{X,U}^{(1)}| = \prod_{i=1}^m a_i.$$

Thus, corollary 2 directly implies that there holds the following corollary.

Corollary 3. Let $U = \{U_i | i \in \mathbf{N}_m\}$ be a set of unitary operators that commute with each other, such that for every $i \in \mathbf{N}_m$ there exists some integer a_i that satisfies identity $U_i^{a_i} = I$. Then any language accepted (either with given probability, or with given mistake) by the model MO-1QFA, as well as by the model L-QFA with measurement at final instant only is union of some elements of the factor-set $X^+ / \equiv_{X,U}^{(1)}$.

Now we consider the situation that takes the place for the models MM-1QFA, k QFA and N-QFA existing in some initial pure state $|\varphi_0\rangle \in \mathbf{C}^n$ under assumption that measurements in the basis $\mathbf{B}_n$ can be applied at intermediate steps. The following two principally different cases are possible.

Case 1. At any intermediate instant the presence of measurement depends only on the current state of investigated model.

Let investigated model exists in state $|\varphi\rangle \in \mathbf{C}^n$ at current instant and it remains to process input substring w . We denote by $w_{|\varphi\rangle}$ the shortest initial

fragment of w that transforms the state $|\varphi\rangle$ in the state $U_{w|\varphi\rangle}|\varphi\rangle$, such that next measurement can be applied. Processing of any input string by investigated model can be presented via marked rooted ranged oriented tree T designed on the base of the following procedure.

Procedure. Let processed vertex v be labeled with some state $|\varphi\rangle \in \mathbf{C}^n$ and $w_{|\varphi\rangle} \neq w$.

The arrow with label $w_{|\varphi\rangle}$ leads to vertex v' of the next level of the tree. The vertex v' is labeled with the state $U_{w_{|\varphi\rangle}}|\varphi\rangle$.

Non-labeled arcs lead from the vertex v' to vertices $v_1, \dots, v_{|Q_{nt}|}, v_{|Q_{nt}|+1}, v_{|Q_{nt}|+2}$ of the next level of the tree. Any vertex v_i ($i \in \mathbf{N}_{|Q_{nt}|}$) is determined to be non-terminal and is labeled by corresponding basic state $q \in Q_{nt}$ and by probability that just this state exists in the result of measurement. Vertices $v_{|Q_{nt}|+1}$ and $v_{|Q_{nt}|+2}$ are determined to be terminal. Vertex $v_{|Q_{nt}|+1}$ (correspondingly, $v_{|Q_{nt}|+2}$) is labeled with the set Q_{acc} (correspondingly, Q_{rej}) and with probability that in the result of measurement investigated model exists in some basic state $q \in Q_{acc}$ (correspondingly, $q \in Q_{rej}$).

Let processed vertex v be labeled with some state $|\varphi\rangle \in \mathbf{C}^n$ and $w_{|\varphi\rangle} = w$.

The arrow with label w leads to vertex v' of the next level of the tree. The vertex v' is labeled with the state $U_w|\varphi\rangle$.

Two non-labeled arcs lead from the vertex v' to vertices v_1 and v_2 of the next level of the tree. These vertices are determined to be terminal. Vertex v_1 (correspondingly, v_2) is labeled with the set Q_{acc} (correspondingly, Q_{rej}) and with probability that in the result of measurement investigated model exists in some basic state $q \in Q_{acc}$ (correspondingly, $q \in Q \setminus Q_{acc}$).

The root of the tree T is labeled with initial state $|\varphi_0\rangle \in \mathbf{C}^n$ and with probability that is equal to 1.

Proposed procedure is applied starting with the root of the tree T till there exists at least one non-terminal vertex.

Let us consider any path π leading from the root of the tree T to terminal vertex labeled with the set Q_{acc} . It is evident that input string that is equal to concatenation of input strings that label arcs along

path π is some initial segment of processed input string. Moreover, this initial segment is accepted by investigated model with probability p_π that is equal to product of probabilities along considered path.

Thus, processed input string is accepted by investigated model at least with probability $P_T = \min_{\pi} p_\pi$, where minimum is over all paths π leading from the root of the tree T to terminal vertices labeled with the set Q_{acc} .

We can substitute associated classes of the factor-set $X^+ / \equiv_{X,U}$ instead of input strings that label arcs of the tree T_w . Thus, with any path π leading from the root of the tree T_w to terminal vertex labeled with the set Q_{acc} it is associated some language L_π that is equal to concatenation of elements of the factor-set $X^+ / \equiv_{X,U}$ along path π .

It is evident that sub-language L_π consists some initial segments of input strings that form the element of the factor-set $X^+ / \equiv_{X,U}$ associated with processed input string. Since sub-language L_π is accepted by investigated model with probability p_π , then sub-language $L_T = \bigcup_{\pi} L_\pi$ (where union is over all paths π leading from the root of the tree T to terminal vertices labeled with the set Q_{acc}) is accepted by investigated model at least with probability $P_{L_T} = \min_{\pi} p_\pi$ (where minimum is over all paths π leading from the root of the tree T to terminal vertices labeled with the set Q_{acc}).

Case 2. At any intermediate instant the presence of measurement depends on observed input symbol and current state of investigated model.

In this case processing of input string also can be presented via marked rooted ranged oriented tree T presented in case 1. Essential difference lay in the factor that associated classes of the factor-set $X^+ / \equiv_{X,U}$ couldn't be substituted instead of input strings that label arcs of the tree T .

Conclusions.

In given paper the sets of finite QA under assumption that unitary operators associated with input alphabet commute each with the others are investigated. Languages accepted by these models either with given probability, or with given mistake are characterized via associated factor-semigroup.

Identities (11) and (14) imply that two pure states $|\varphi_0\rangle, |\tilde{\varphi}_0\rangle \in \mathbf{C}^n$ of the model MO-1QFA are:

1) equivalent with respect to given probability p ($0.5 < p \leq 1$) if and only if it holds identity

$$L_M(|\varphi_0\rangle, p) = L_M(|\tilde{\varphi}_0\rangle, p);$$

2) equivalent with respect to given mistake (p_1, p_2) ($0 \leq p_1 < p_2 < 1$) if and only if it holds identity

$$L_M(|\varphi_0\rangle, (p_1, p_2)) = L_M(|\tilde{\varphi}_0\rangle, (p_1, p_2)).$$

Similarly, identities (17) and (18) imply that two mixed states $\{(|\psi_i\rangle, p_i)\}_{i \in N_I}$ and $\{(|\tilde{\psi}_i\rangle, p_i)\}_{i \in N_I}$ of the model L-QFA with measurement at final instant only are:

1) equivalent relatively to given probability p ($0.5 < p \leq 1$) if and only if it holds identity

$$L_M(\{(|\psi_i\rangle, p_i)\}_{i \in N_I}, p) = L_M(\{(|\tilde{\psi}_i\rangle, p_i)\}_{i \in N_I}, p);$$

2) equivalent relatively to given mistake $(\tilde{p}_1, \tilde{p}_2)$ ($0 \leq \tilde{p}_1 < \tilde{p}_2 < 1$) if and only if it holds identity

$$\begin{aligned} L_M(\{(|\psi_i\rangle, p_i)\}_{i \in N_I}, (\tilde{p}_1, \tilde{p}_2)) = \\ = L_M(\{(|\tilde{\psi}_i\rangle, p_i)\}_{i \in N_I}, (\tilde{p}_1, \tilde{p}_2)). \end{aligned}$$

Список використаних джерел

1. Moore C., Crutchfield J. Quantum automata and quantum grammars // Theor. Comput. Sci. – 2000. – Vol. 237. – P. 257-306.
2. Brodsky A., Pipenger N. Characterizations of 1-way quantum finite automata // Journal of Computing. – 2002. – № 5. – P. 1456-1478.
3. Kondacs A., Watrous J. On the power of quantum finite state automata // Proceedings of the 38th IEEE Symposium on Foundations of Computer Science. – 1997. – P. 66-75.
4. Ambainis A., Kikusts A., Valdat M. On some class of languages recognizable by 1-way quantum automata // LNCS. – 2001. – Vol. 2010. – P. 75-86.
5. Ambainis A., Beaudry M., Golovkins M., *at all*. Algebraic results on quantum automata // LNCS. – 2004. – Vol. 2996. – P. 93-104.
6. Nayak A. Optimal lower bounds for quantum automata and random access codes // Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science. – 1999. – P. 369-377.
7. Belovs A., Rosmanis A., Smotrovs J. Multi-letter reversible and quantum finite automata // LNCS. – 2007. – Vol. 4588. – P. 60-71.
8. Bertoni A., Mereghetti C., Palano B. Quantum computing: 1-way quantum automata // LNCS. – 2003. – Vol. 2710. – P. 1-20.
9. Amano V., Iwama K. Undecidability on quantum finite automata // Proceedings of the 31st Annual ACM Symposium on Theory of Computation. – 1999. – P. 368-375.
10. Dubrovsky A. Space-efficient 1.5-way Turing machine // LNCS. – 2001. – Vol. 2138. – P. 380-383.
11. Ciamarra M.P. Quantum reversibility and a new model of quantum automaton // LNCS. – 2001. – Vol. 2138. – P. 376-379.
12. Kravtsevs M. Quantum one counter automata // LNCS. – 1999. – Vol. 1725. – P. 431-440.
13. Yamasali T., Kobayashi H., Tokunaga Y., *at all*. One-way probabilistic reversible and quantum automata // LNCS. – 2000. – Vol. 1858. – P. 436-446.
14. Bonner R., Freivalds R., Kravtsevs M. Quantum versus probabilistic one-way finite automata with counter // LNCS. – 2001. – Vol. 2234. – P. 181-190.
15. Yamasali T., Kobayashi H., Imai H. Quantum versus deterministic counter automata // LNCS. – 2002. – Vol. 2387. – P. 584-594.

Надійшла до редколегії 29.01.11

УДК 532.59

Авраменко О.В., д.ф.-м.н., проф.
Гуртовий Ю.В., к.ф.-м.н., доц.
Нарадовий В.В.

Оцінка енергії хвильового руху у двошаровій рідині з твердим дном та вільною поверхнею

Досліджується енергія хвильового руху у двошаровій рідині скінченної глибини. Виявлено залежності енергії поверхневого та внутрішнього хвильового руху від геометричних та фізичних параметрів гідродинамічної системи «шар з твердим дном – шар з вільною поверхнею». Знайдено такі параметри системи, для яких енергія набуває екстремальних значень. Всі результати проілюстровані графічно.

Ключові слова: енергія хвилі, внутрішня хвиля, поверхнева хвиля.

Кіровоградський державний педагогічний університет імені Володимира Винниченка, 25001, м. Кіровоград, вул. Шевченка, 1,

e-mail: oavramenko@rambler.ru
gyrtovuy@rambler.ru
naradvova@rambler.ru

Avramenko O.V., Dr. Phys.-Math. Sci., Prof.
Hurtovyy Yu.V., Cand. Phys.-Math. Sci., Doc.
Naradovyy V.V.

Estimation of wave motion energy in the two-layer fluid with solid bottom and free surface

The energy of the wave motion in the two-layer fluid of finite depth is investigated. Dependence of energy of the surface and internal waves on the geometrical and physical parameters of the hydrodynamic system of "layer with solid bottom - layer with free surface" is found. The parameters of the system for which the energy takes extreme values are found. All results are illustrated graphically.

Key Words: wave energy, internal wave, surface wave

Kirovohrad Volodymyr Vynnychenko State Pedagogical University, 25001, Kirovohrad, Shevchenko str., 1,

e-mail: oavramenko@rambler.ru
gyrtovuy@rambler.ru
naradvova@rambler.ru

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

Вступ

Вивчення фізичних процесів у стратифікованих гідродинамічних структурах, а саме дослідження слабконелінійних хвильових пакетів у двошаровій рідкій системі з дисперсією та якісного і кількісного аналізу їх поширення, еволюції та стійкості в залежності від фізичних та геометричних параметрів рідин, є однією з актуальних проблем гідромеханіки. Актуальність проблематики зумовлена тим, що внутрішні та поверхневі хвильові рухи відіграють важливу роль у динамічних явищах у Світовому океані, що викликає їх спостереження у натурних умовах та різні теоретичні та експериментальні дослідження.

Черкесов у роботах [1, 2] провів дослідження поширення внутрішніх та поверхневих хвиль в однорідній та неоднорідній системах у залежності від ряду параметрів вказаних систем у

наближенні мілкої води. Ним було досліджено у лінійному наближенні вплив в'язкості на рух довгих хвиль у двошаровій рідині скінченної глибини з вільною поверхнею та з кришкою. Проаналізовано вплив капілярності та періодичних збурюючих тисків на деформацію просторових поверхневих хвиль. Досліджено також рух та деформацію хвиль цунамі у прибережних зонах.

У монографії [3] розглянуто теоретичні та експериментальні дослідження поширення внутрішніх та поверхневих хвиль для однорідних та неоднорідних рідин. Доведені теореми існування та єдиності розв'язків задачі Коші – Пуассона, а також наведені обґрунтування наближених теорій – лінійної та мілкої води. Зокрема, Овсянніковим розглянуті лінійні задачі поширення хвиль у системах «шар з вільною поверхнею», «шар з твердим дном – шар з

вільною поверхнею», «шар з твердим дном – шар з кришкою». Також наведений вивід ряду математичних моделей другого наближення за Лагранжем для різних типів хвильового руху. Отримані розв'язки другого наближення за Лагранжем для системи «шар з твердим дном – шар з кришкою», а також проаналізовані умови проходження хвиль в системі «шар з твердим дном – шар з вільною поверхнею» на основі аналізу дисперсійного співвідношення.

Постановка задачі та її розв'язки

Досліджується слабконелінійна задача про поширення хвиль на поверхні нижнього рідкого шару з твердим дном $\Omega_1 = \{(x, z) : |x| < \infty, -h_1 \leq z < 0\}$ та верхнього рідкого шару з вільною поверхнею $\Omega_2 = \{(x, z) : |x| < \infty, 0 \leq z \leq h_2\}$. Густини рідин дорівнюють ρ_1 та ρ_2 , відповідно. Шари розділені поверхнею контакту $z = \eta(x, t)$, а верхній шар обмежений згори вільною поверхнею $z = \eta_0(x, t)$. При розв'язанні враховується сила поверхневого натягу на поверхні контакту та на вільній поверхні. Сила тяжіння направлена у від'ємному z -напрямку, рідини нестисливі. Постановка задачі

$$\begin{aligned} \nabla^2 \varphi_j &= 0 \quad \text{в } \Omega_j, \\ \eta_{,t} - \varphi_{j,z} &= -\varphi_{j,x} \eta_{,x} \quad \text{при } z = \eta(x, t), \\ \eta_{0,t} - \varphi_{2,z} &= -\varphi_{2,x} \eta_{0,x} \quad \text{при } z = \eta_0(x, t), \\ \varphi_{1,t} - \rho \varphi_{2,t} + (1 - \rho) \eta &+ 0.5 \alpha \left[(\nabla \varphi_1)^2 - \rho (\nabla \varphi_2)^2 \right] - \\ &- T (1 + \alpha^2 \eta_{,x}^2)^{-3/2} \eta_{,xx} = 0 \quad \text{при } z = \eta(x, t), \end{aligned} \quad (1)$$

для ω_1 :

$$\eta_1^{(1)} = A \sin(kx - \omega_1 t),$$

$$\eta_{01}^{(1)} = \frac{\omega_1^2}{\omega_1^2 \operatorname{ch}(kh_2) - (k + T_0 k^3) \operatorname{sh}(kh_2)} A \sin(kx - \omega_1 t),$$

$$\varphi_{11}^{(1)} = -\frac{A}{\omega_1 \operatorname{ch}(kh_1)} \operatorname{ch}(k(h_1 + z)) \cos(kx - \omega_1 t), \quad (3)$$

$$\varphi_{21}^{(1)} = -\frac{A \omega_1 \operatorname{th}(kh_1) \left(\omega_1^2 \operatorname{sh}(k(h_2 - z)) - (k + T_0 k^3) \operatorname{ch}(k(h_2 - z)) \right)}{\left(\omega_1^2 \operatorname{ch}(kh_2) - (k + T_0 k^3) \operatorname{sh}(kh_2) \right)^2} \cos(kx - \omega_1 t).$$

для ω_2 :

$$\eta_1^{(2)} = \frac{\omega_2^2 \operatorname{ch}(kh_2) - (k + T_0 k^3) \operatorname{sh}(kh_2)}{\omega_2^2} B \sin(kx - \omega_2 t),$$

$$\eta_{01}^{(2)} = B \sin(kx - \omega_2 t),$$

$$\varphi_{11}^{(2)} = -\frac{\omega_2^2 \operatorname{ch}(kh_2) - (k + T_0 k^3) \operatorname{sh}(kh_2)}{\omega_2^3 \operatorname{ch}(kh_1)} B \operatorname{ch}(k(h_1 + z)) \cos(kx - \omega_2 t), \quad (4)$$

$$\varphi_{21}^{(2)} = -\frac{B \operatorname{th}(kh_1) \left(\omega_2^2 \operatorname{sh}(k(h_2 - z)) - (k + T_0 k^3) \operatorname{ch}(k(h_2 - z)) \right)}{\omega_2 \left(\omega_2^2 \operatorname{ch}(kh_2) - (k + T_0 k^3) \operatorname{sh}(kh_2) \right)} \cos(kx - \omega_2 t),$$

$$\varphi_{2,t} + \eta_0 + 0.5 \alpha (\nabla \varphi_2)^2 - T_0 \left(1 + \alpha^2 \eta_{0,x}^2 \right)^{-3/2} \eta_{0,xx} = 0,$$

$$\varphi_{1,z} = 0 \quad \text{при } z = -h_1.$$

Для розв'язку задачі (1) використаний метод багатомасштабних розвинень до третього порядку

$$\eta(x, t) = \sum_{n=1}^3 \alpha^{n-1} \eta_n(x_0, x_1, x_2, t_0, t_1, t_2) + O(\alpha^3), \quad (2)$$

$$\eta_0(x, t) = \sum_{n=1}^3 \alpha^{n-1} \eta_{0n}(x_0, x_1, x_2, t_0, t_1, t_2) + O(\alpha^3),$$

$$\varphi_j(x, z, t) = \sum_{n=1}^3 \alpha^{n-1} \varphi_{jn}(x_0, x_1, x_2, z, t_0, t_1, t_2) + O(\alpha^3),$$

де масштаби $x_j = \alpha^j x$ та $t_j = \alpha^j t$ ($j=0, 1, 2$), а α – коефіцієнт нелінійності

Підстановка розвинень (2) у постановку нелінійної задачі (1) та прирівнювання виразів при однакових степенях α призводить до трьох перших наближень проблеми.

Детальний аналіз розв'язків перших двох наближень наведений у [4, 6-10]. Із першого лінійного наближення було отримано дисперсійне співвідношення, яке пов'язує між собою хвильове число та частоту хвиль.

Вперше серед задач цього класу дисперсійне співвідношення має дві пари коренів $\pm \omega_1$ та $\pm \omega_2$, які відповідають поверхневим хвилям на вільній поверхні та внутрішнім хвилям на поверхні контакту. Для кожної з пар частот можна записати незалежні розв'язки першої лінійної проблеми

де A - амплітуда внутрішньої прогресивної хвилі $\eta_1^{(1)}$, B - амплітуда поверхневої прогресивної хвилі $\eta_{01}^{(2)}$. Тут $\eta_{01}^{(1)}$ є поверхневою хвилею-відгуком на внутрішню хвилю $\eta_1^{(1)}$ з частотою ω_1 , а $\eta_1^{(2)}$ є внутрішньою хвилею-відгуком на поверхневу хвилю $\eta_{01}^{(2)}$ з частотою ω_2 .

Дослідження енергії хвильового руху у першому наближенні

Розглянемо енергію, яку переносять через площину $x = \text{const}$ за період τ двовимірні прогресивні хвилі, що задаються розв'язками (3) та (4). Згідно [5] мають місце формули

$$E_{vn} = -\frac{\rho}{\tau} \int_t^{t+\tau} dt \int_{-h_1}^0 \frac{\partial(\varphi_{11}^{(1)} + \varphi_{11}^{(2)})}{\partial t} \frac{\partial(\varphi_{11}^{(1)} + \varphi_{11}^{(2)})}{\partial x} dz, \quad (5)$$

$$E_{pov} = -\frac{1}{\tau} \int_t^{t+\tau} dt \int_0^{h_2} \frac{\partial(\varphi_{21}^{(1)} + \varphi_{21}^{(2)})}{\partial t} \frac{\partial(\varphi_{21}^{(1)} + \varphi_{21}^{(2)})}{\partial x} dz, \quad (6)$$

де E_{vn} - енергія, яку переносить внутрішня хвиля, E_{pov} - енергія, яку переносить поверхнева хвиля.

Отримані результати доцільно порівняти з розрахунком енергії, що переносить хвиля у гідродинамічній системі «шар з вільною поверхнею» [5]

$$\zeta = C \sin(\omega t - kx),$$

$$\varphi = \frac{C}{\omega \operatorname{ch}(kh)} \operatorname{ch}(k(z+h)) \cos(\omega t - kx).$$

$$E_{kl} = -\frac{\rho}{\tau} \int_t^{t+\tau} dt \int_{-h}^0 \frac{\partial \varphi}{\partial t} \frac{\partial \varphi}{\partial x} dz. \quad (7)$$

Дослідимо залежності від товщини нижнього шару h_1 енергії хвильових рухів для різних геометричних та фізичних параметрів: 1) випадок рівності густин рідких шарів $\rho_1 = \rho_2$, $\rho = 1$ (однорідний півпростір); 2) випадок поширення прогресивної хвилі тільки на поверхні контакту шарів з різними густинами; 3) випадок поширення прогресивної хвилі тільки на вільній поверхні двошарової рідини; 4) одночасне поширення внутрішньої та поверхневої хвилі.

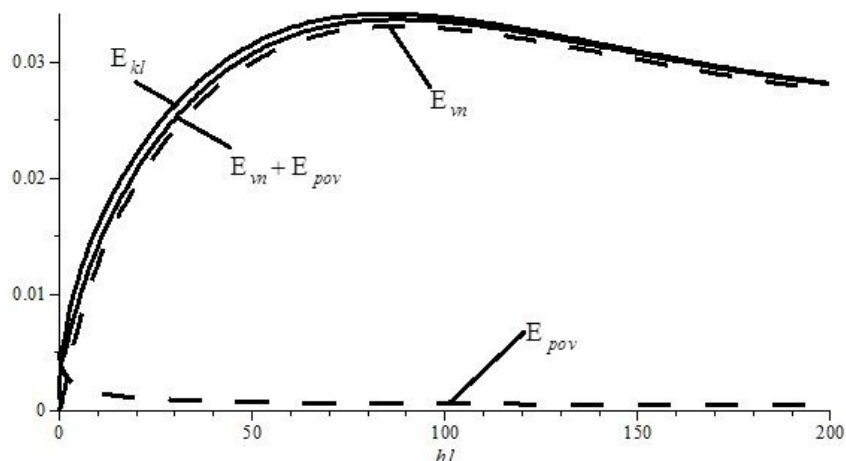


Рис. 1 Залежність E_{vn} , E_{pov} , $E_{vn} + E_{pov}$ та E_{kl} від товщини нижнього шару h_1

1). *Випадок рівності густин* $\rho_1 = \rho_2$, $\rho = 1$. У цьому випадку товщина однорідного шару $h = h_1 + h_2$ та амплітуда внутрішньої прогресивної хвилі $A = 0$. Шар обмежений згори вільною поверхнею. На рис.1 показана залежність E_{vn} , E_{pov} , $E_{vn} + E_{pov}$ та E_{kl} від товщини нижнього шару для наступних параметрів $k = 0.01$, $T = T_0 = 0$, $h_2 = 1$, амплітуда поверхневої прогресивної хвилі $B = 0.1$.

Як помітно з рис.1 при збільшенні товщини нижнього шару енергія, що переноситься поверхневою хвилею, спадає. При цьому сума

внутрішньої та поверхневої енергії (енергія системи) $E_{vn} + E_{pov}$ наближається до енергії E_{kl} однорідного шару товщини $h = h_1 + h_2$. Отже, для довгих хвиль малої амплітуди сума енергій, що переносяться внутрішньою та поверхневою хвилями, наближається до енергії, що відповідає поверхневій хвилі у півпросторі.

2). *Поширення внутрішньої прогресивної хвилі з амплітудою $A \neq 0$ у двошаровій системі.*

На рис.2 представлено залежність енергії, яку переносить внутрішня хвиля з амплітудою A на поверхні контакту та її відгук на вільній поверхні, від товщини нижнього шару h_1 при

різних хвильових числах $k = 0.01$, $k = 0.1$, $k = 1$, $k = 2$ для наступних параметрів системи $h_2 = 1$, $\rho = 0.95$, $T = T_0 = 0$, амплітуда внутрішньої хвилі $A = 0.1$, амплітуда поверхневої хвилі $B = 0$.

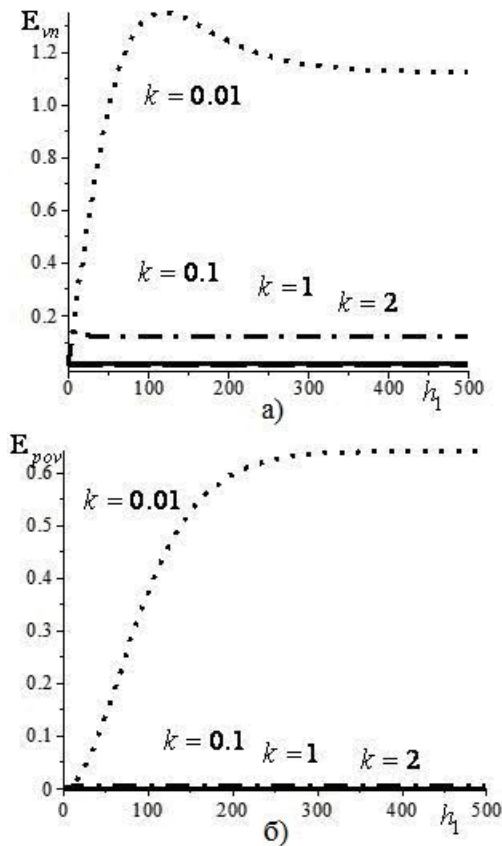


Рис. 2. Залежність від товщини нижнього шару:
(а) енергія внутрішньої хвилі;
(б) енергія відгуку внутрішньої хвилі

Видно (рис.2), що енергія внутрішньої хвилі має найбільше значення для певних товщин нижнього шару, при збільшенні товщини h_1 її величина прямує до сталої величини. При збільшенні хвильового числа величина енергії зменшується, екстремум досягається при менших товщинах нижнього шару. На вільній поверхні з'являється поверхнева хвиля-відгук на збурення на поверхні контакту. Її енергія зростає зі зростанням товщини нижнього шару h_1 , при цьому екстремум не виявлений. Для невеликих товщин нижнього шару енергія відгуку на два порядки менша за енергію внутрішньої хвилі.

3). *Поширення поверхневої прогресивної хвилі з амплітудою $B \neq 0$ у двошаровій системі.*

На рис.3 представлено залежність енергії, яку переносить поверхнева хвиля з амплітудою B на поверхні контакту та її відгук на вільній

поверхні, від товщини нижнього шару h_1 при різних хвильових числах $k = 0.01$, $k = 0.1$, $k = 1$, $k = 2$ для наступних параметрів системи $h_2 = 1$, $\rho = 0.95$, $T = T_0 = 0$, амплітуда внутрішньої хвилі $A = 0$, амплітуда поверхневої хвилі $B = 0.05$.

Енергія внутрішньої хвилі при $h_1 = 0$ відповідає значенню енергії хвилі у однорідному шарі товщиною $h_2 = 1$, при збільшенні h_1 прямує до фіксованого значення: для великих хвильових чисел енергія поверхневої хвилі зростає, а для малих – спадає. Енергія внутрішнього відгуку на поверхневе збурення (рис.3) досягає екстремальних значень для певних товщин нижнього шару.

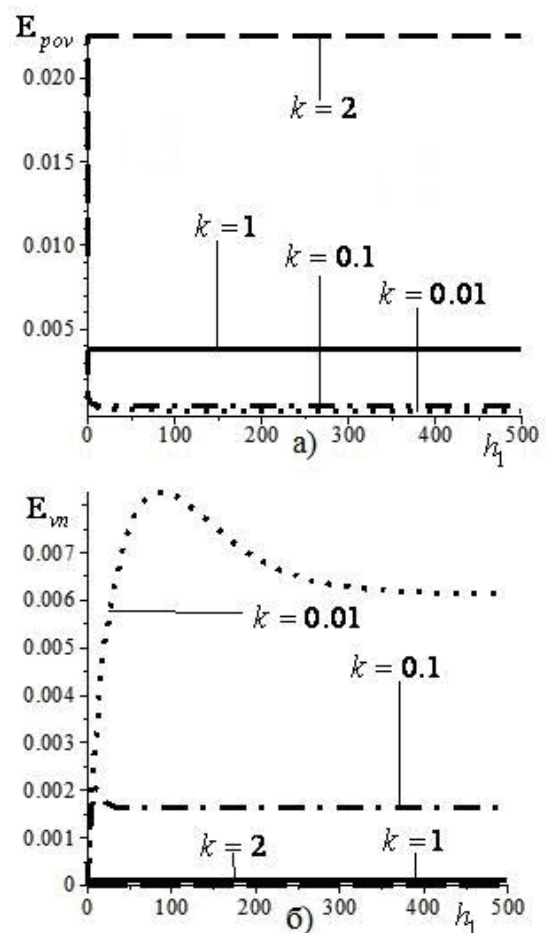


Рис.3. Залежність від товщини нижнього шару:
(а) енергія поверхневої хвилі;
(б) енергія відгуку поверхневої хвилі

4). *Поширення внутрішньої та поверхневої прогресивних хвиль з амплітудами $A \neq 0$, $B \neq 0$ у двошаровій системі.*

За формулами (5) та (6) обчислено енергію, яку переносять внутрішня та поверхнева хвилі, при цьому відхилення поверхні контакту та

вільної поверхні задаються як $\eta_1 = \eta_1^{(1)} + \eta_1^{(2)}$,
 $\eta_{01} = \eta_{01}^{(1)} + \eta_{01}^{(2)}$.

На рисунках нижче показано залежність E_{vn} (рис.4), E_{pov} (рис.5) та їх суми $E_{vn} + E_{pov}$ (рис.6) від хвильового числа k при різних товщинах нижнього шару $h_1 = 10$, $h_1 = 3$, $h_1 = 1$ для таких параметрів системи $h_2 = 1$, $A = 0.1$, $B = 0.05$, $\rho = 0.95$, $T = T_0 = 0$. За умови збільшення

товщини нижнього шару енергія, яку переносить довга ($k \ll 1$) внутрішня хвиля, зростає (рис.4), при цьому енергія внутрішньої хвилі E_{vn} набуває максимальних значень. Якщо ж $k > 1$, то енергія, яку переносить внутрішня хвиля, зростає зі зменшенням товщини нижнього шару h_1 .

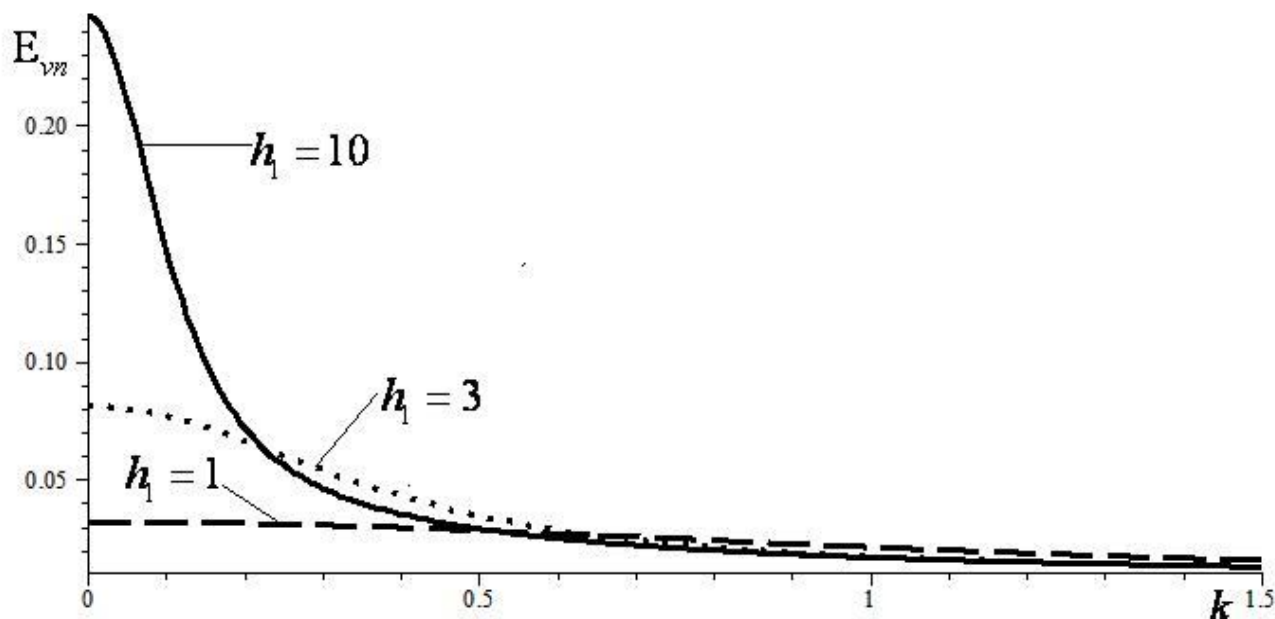


Рис.4. Залежність E_{vn} від k

На рис.5 показано залежність енергії E_{pov} поверхневої хвилі від хвильового числа k . Зі збільшенням товщини нижнього шару, значення E_{pov} зростає, але на відміну від енергії E_{vn} внутрішньої хвилі зміна товщини h_1 не так сильно впливає на значення E_{pov} , зокрема, на рис.5 значення енергії при товщинах $h_1 = 10$ та $h_1 = 3$ є близькими. Зі зростанням хвильового числа енергія E_{pov} , яку переносять поверхневі хвилі, зростає при $0 < k < 1.5$.

На рис.6 показано залежність сумарної енергії $E_{vn} + E_{pov}$ системи від хвильового числа k при $h_1 = 10$, $h_1 = 3$, $h_1 = 1$. Відмітимо, що зміна товщини нижнього шару майже не впливає на величину $E_{vn} + E_{pov}$. Далі розглянемо залежність E_{vn} (рис.7), E_{pov} (рис.8) та $E_{vn} + E_{pov}$ (рис.9) від товщини нижнього шару для різних хвильових

чисел $k = 0.01$, $k = 0.1$, $k = 1$, $k = 2$ при наступних параметрах системи $h_2 = 1$, $A = 0.1$, $B = 0.05$, $\rho = 0.95$, $T = T_0 = 0$.

На рис.7 показано залежність енергії E_{vn} , яку переносить хвиля на поверхні контакту, від товщини нижнього шару. Максимальне значення енергії, як відмічалось вище, набувається при проходженні довгих хвиль при $k = 0.01$.

На рис.8 показано залежність енергії E_{pov} , яку переносить поверхнева хвиля, від товщини нижнього шару. Максимальну енергію для малих товщин нижнього шару переносить коротка поверхнева хвиля $k = 2$. Для довгих хвиль ($k = 0.01$ і $k = 0.1$) спостерігається зменшення енергії поверхневого руху при близьких значеннях товщин шарів системи $h_1 \approx h_2 = 1$, що свідчить про явище гасіння енергії хвиль, яке викликано взаємодією поверхневої прогресивної хвилі з амплітудою B та поверхневого відгуку на внутрішню прогресивну хвилю з амплітудою A .

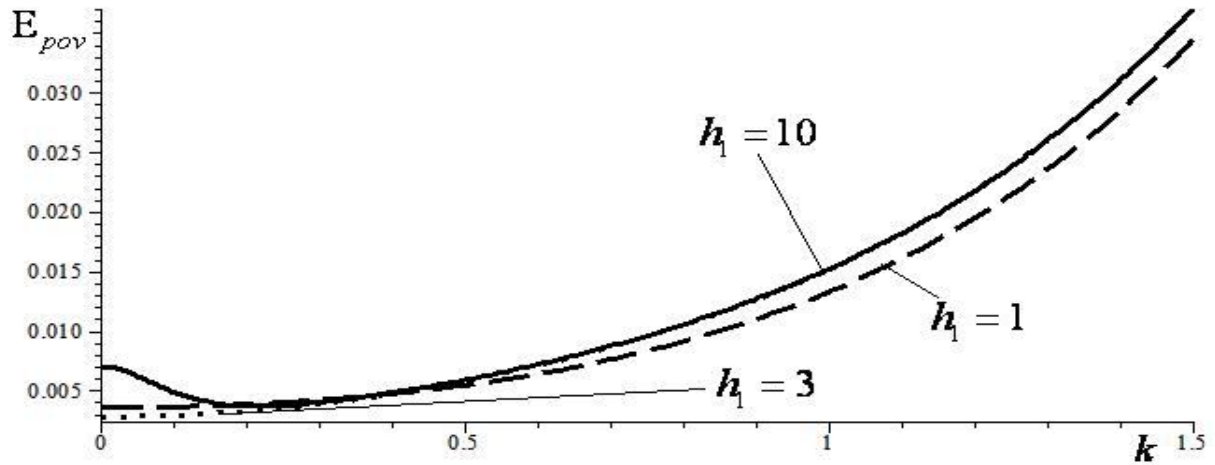


Рис.5. Залежність E_{pov} від k

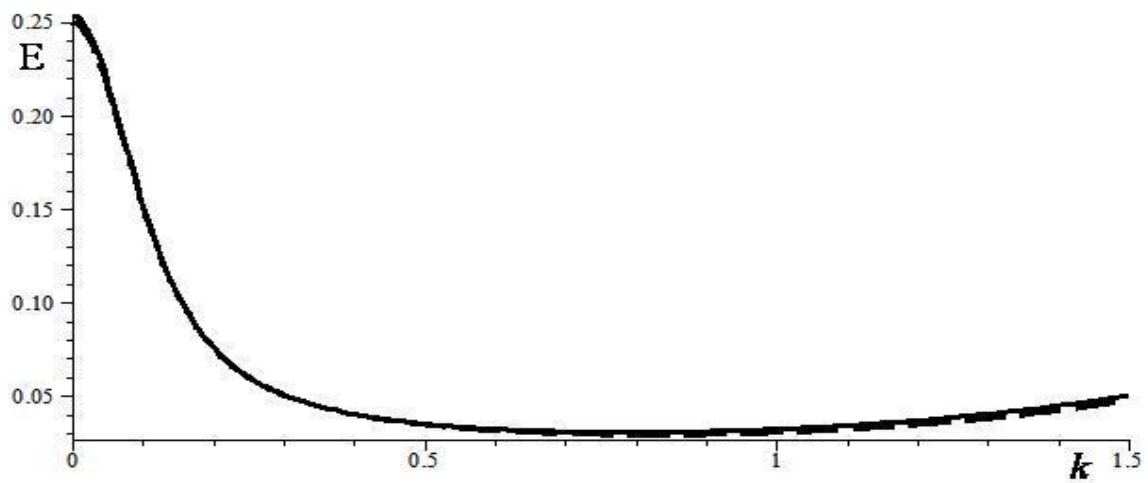


Рис.6. Залежність сумарної енергії $E_{vn} + E_{pov}$ від k

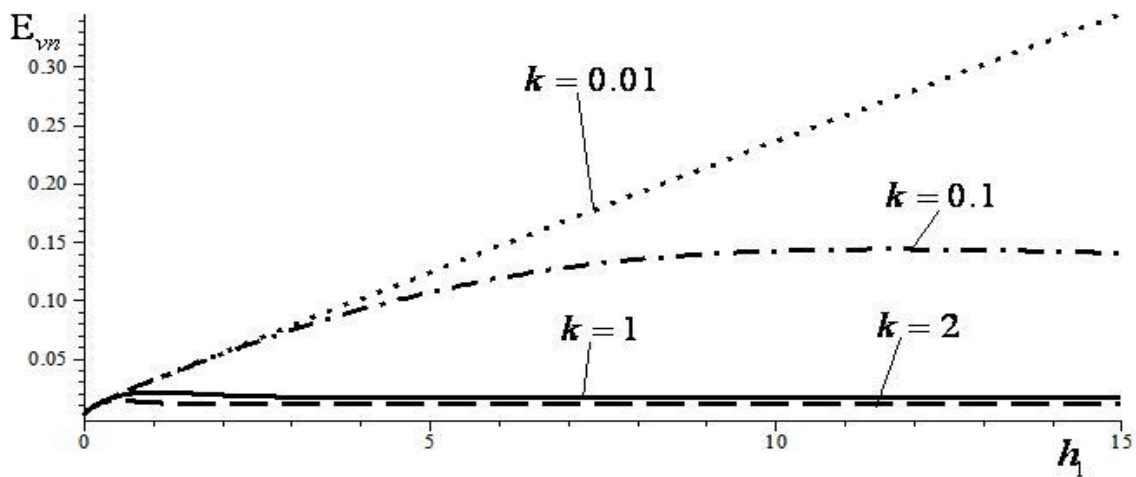


Рис.7. Залежність E_{vn} від h_1

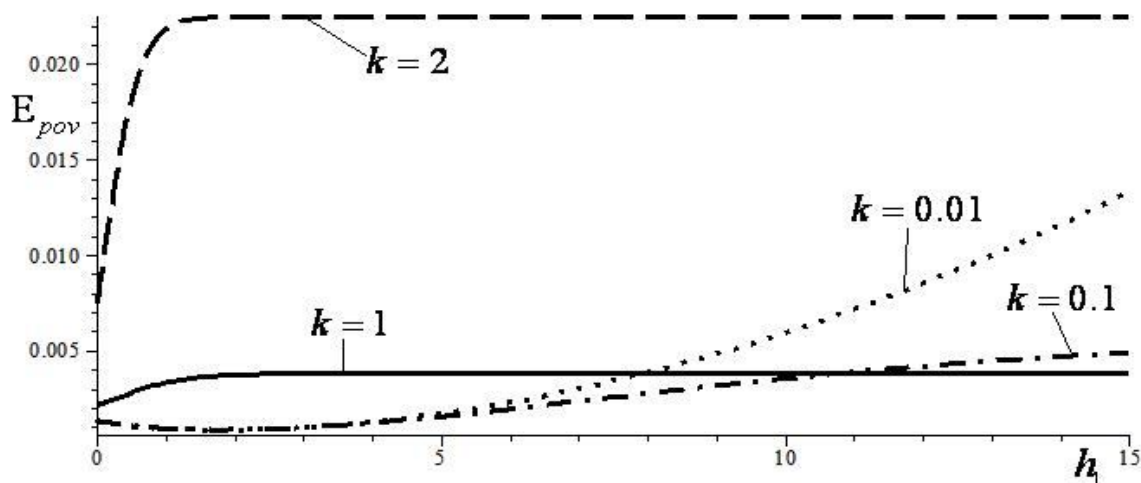


Рис.8. Залежність E_{pov} від h_1

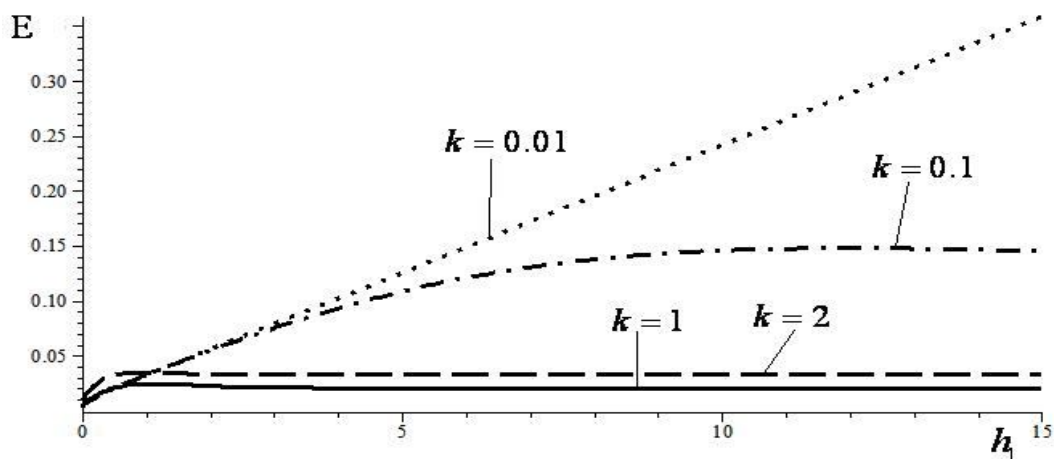


Рис.9. Залежність $E_{vn} + E_{pov}$ від h_1

На рис.9 показано залежність $E_{vn} + E_{pov}$ від h_1 для хвиль різної довжини. Максимальна енергія системи досягається при проходженні на границі контакту довгих внутрішніх хвиль при $h_1 \gg 1$. У випадку $h_1 \ll 1$ енергія системи буде максимальною для великих хвильових чисел, тобто при проходженні на поверхні контакту коротких хвиль (рис.9).

Дослідження енергії хвильового руху у другому наближенні

Вирази для потенціалів у верхньому та нижньому шарах та відхилень вільної поверхні та поверхні контакту у другому наближенні:

$$\begin{aligned} &\text{для } \omega_1 \\ \eta^{(1)} &= \eta_1^{(1)} + \alpha \eta_2^{(1)}, \quad \eta_0^{(1)} = \eta_{01}^{(1)} + \alpha \eta_{02}^{(1)}, \\ \varphi_1^{(1)} &= \varphi_{11}^{(1)} + \alpha \varphi_{12}^{(1)}, \quad \varphi_2^{(1)} = \varphi_{21}^{(1)} + \alpha \varphi_{22}^{(1)}, \\ &\text{для } \omega_2 \end{aligned} \quad (8)$$

$$\eta^{(2)} = \eta_1^{(2)} + \alpha \eta_2^{(2)}, \quad \eta_0^{(2)} = \eta_{01}^{(2)} + \alpha \eta_{02}^{(2)}, \quad (9)$$

$$\varphi_1^{(2)} = \varphi_{11}^{(2)} + \alpha \varphi_{12}^{(2)}, \quad \varphi_2^{(2)} = \varphi_{21}^{(2)} + \alpha \varphi_{22}^{(2)},$$

де $\eta_0^{(1)}$ є поверхневою хвилею-відгуком на внутрішню хвилю $\eta^{(1)}$ з частотою ω_1 . Відповідно $\eta^{(2)}$ є внутрішньою хвилею-відгуком на поверхневу хвилю $\eta_0^{(2)}$ з частотою ω_2 .

Формули для обчислення енергії з урахуванням другого наближення:

$$E_{vn} = -\frac{\rho}{\tau} \int_t^{t+\tau} dt \int_{-h_1}^{\eta_1} \frac{\partial(\varphi_1^{(1)} + \varphi_1^{(2)})}{\partial t} \frac{\partial(\varphi_1^{(1)} + \varphi_1^{(2)})}{\partial x} dz, \quad (10)$$

$$E_{pov} = -\frac{1}{\tau} \int_t^{t+\tau} dt \int_{\eta_1}^{\eta_{01}} \frac{\partial(\varphi_2^{(1)} + \varphi_2^{(2)})}{\partial t} \frac{\partial(\varphi_2^{(1)} + \varphi_2^{(2)})}{\partial x} dz, \quad (11)$$

де E_{vn} - енергія, яку переносить внутрішня хвиля $\eta = \eta^{(1)} + \eta^{(2)}$, E_{pov} - енергія, яку

переносить поверхнева хвиля $\eta_0 = \eta_0^{(1)} + \eta_0^{(2)}$. Відмітимо, що у формулах (10) та (11) межі інтегрування по z вже не є сталими величинами. У (10) інтегрування по z здійснюється в межах від 0 до $\eta_1 = \eta_1^{(1)} + \eta_1^{(2)}$, а в (11) – від $\eta_1 = \eta_1^{(1)} + \eta_1^{(2)}$ до $\eta_{01} = \eta_{01}^{(1)} + \eta_{01}^{(2)}$, що обумовлено врахуванням другого наближення. Оцінку внеску другого наближення в повну енергію системи визначено як відношення квадратів амплітуд других гармонік, які задаються виразами (8)–(9). Розрахунок здійснено для різних хвильових чисел $k = 0.01$, $k = 0.1$, $k = 1$, $k = 2$ та відношень густин $\rho = 0.9$, $\rho = 0.95$, $\rho = 0.99$ та для наступних параметрів системи $h_1 = 10$, $h_2 = 1$, $T = T_0 = 0$, амплітуда внутрішньої хвилі $A = 0.1$, амплітуда поверхневої хвилі $B = 0.05$. В результаті отримано, що цей внесок складає 3% від загальної енергії хвильового руху, що можна пояснити тим, що розв'язки другого наближення входять в загальні розвинення для потенціалів та відхилень поверхні контакту та вільної поверхні (2) з коефіцієнтом нелінійності α , який є достатньо малим. Отже, якщо розглядати ті параметри системи, за яких не спостерігається явище резонансу другої гармоніки, можна

обмежитись лише дослідженням енергії в лінійному наближенні.

Відмітимо, що для виконання громіздких символічних перетворень, чисельних розрахунків і контролю достовірності результатів були розроблені програмні засоби у середовищі пакету комп'ютерної алгебри Maple [11].

Висновки

В роботі досліджується енергія хвильового руху в гідродинамічній системі «шар з твердим дном – шар з вільною поверхнею». У випадку неоднорідної рідини виявлено, що максимальну енергію на поверхні контакту переносять довгі внутрішні хвилі, а максимальну енергію на вільній поверхні – короткі поверхневі хвилі. Відповідні результати проілюстровано графічно. Виконано оцінку внеску другого наближення в загальну енергію системи. Внесок другого наближення у загальну енергію системи складає близько 3% оскільки друге наближення входить в загальний розв'язок з коефіцієнтом нелінійності, який є порівняно малим порівняно з амплітудою першого наближення.

Фізична достовірність отриманих результатів підтверджена граничними переходами у вироджених випадках за умови перетворення двошарової системи в одношарову.

Список використаних джерел

1. *Cherkesov L.V.* Hydrodynamics of surface and internal waves.- Kiev: Naukova Dumka, 1976.- 364 p. (In Russian)
2. *Cherkesov L.V.* Surface and internal waves.- Kiev: Naukova Dumka, 1973.- 248 p. (In Russian)
3. *Ovsiannikov V.L., Makarenko N.I., Nalimov V.I.* Nonlinear problems of the theory of surface and internal waves.- Novosibirsk: Nauka, 1985.- 318 p. (In Russian)
4. *Selezov I.T., Krivonos Y.G.* Mathematical methods of propagation and diffraction of waves.- Kiev: Naukova Dumka, 2012.- 232 p. (In Russian)
5. *Tarapov I.E.* Mechanics of continua. In 3 parts. P. 3: Mechanics of inviscid fluid.- Harkiv: Golden Pages, 2005.- 332 c. (In Russian)
6. *Avramenko O.V., Naradovyy V.V., Selezov I.T.* Multyscale modelling of the wave interaction in two-layer fluid with free surface // Proceedings of the Institute of Mathematics of NAS of Ukraine, 2010. – Vol.7, №2.- P. 2-9.

7. *Avramenko O., Naradovyy V.* Stability of wave-packets in the two-layer fluid with free surface and rigid bottom // Contemporary problems of mathematics, mechanics and computing sciences.- Harkiv, 2012. – Vol. 1.

8. *Avramenko O., Naradovyy V.* Multiscale approximation of surface and internal wave-packets interaction in two-layer fluid // Contemporary problems of mathematics, mechanics and computing sciences.-Harkiv, 2013. – Vol. 2.

9. *Selezov I.T., Avramenko O.V., Gurtovyy Y.V., Naradovyy V.V.* Nonlinear interaction of internal and surface gravity waves in a two-layer fluid with free surface // J. of Math. Science.- 2010.- Vol.168, №4.- P. 590 -602.

10. *Selezov I.T., Avramenko O.V., Naradovyy V.V.* Some features of nonlinear wave propagating in two-layer fluid with free surface // Dynamical Systems.- 2011.- Vol. 1(29), №1.-P. 53-68. (In Russian)

11. *Lynch S.* Dynamical Systems with Applications using Maple.-2010.-234 p.

Надійшло до редколегії 15.07.2013

УДК 004.655

Буй Д.Б.¹, д.ф.-м.н., проф.,
Кахута Н.Д.², к.ф.-м.н.,
Сільвейструк Л.М.³, к.ф.-м.н., н.с.

Загальна теорія бінарних відношень: проекція та відношення сумісності

Метою статті є дослідження теоретико-множинних конструкцій для подальшого використання в табличних алгебрах. В роботі розглянуто проекцію та відношення сумісності та доведена низка критеріїв та властивостей даних конструкцій.

Ключові слова: проекція, відношення сумісності, таблична алгебра, повний образ.

¹Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Академіка Глушкова 4д, e-mail: dmitriybuy@mail.ru

²Університет економіки та права «КРОК», 03113, м. Київ, вул. Лагерна 30-32, e-mail: krok@krok.edu.ua

³Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Академіка Глушкова 4д, e-mail: slm-klm@ukr.net

D.B. Buy¹, Dr. Sci. (Phys.-Math.), Prof.
N.D. Kahuta², PhD (Phys.-Math.),
L.M. Silvestruk³, PhD. (Phys.-Math.), Sci. Res.

The general theory of binary relations: the projection and related compatibility

The purpose of the article is to study the set-theoretic constructions for use in tabular algebra. The projection and compatibility relation are viewed and criteria and the properties of these structures are proved.

Key Words: projection, compatibility relation, tabular algebra, total transform.

¹Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova st., 4d
e-mail: dmitriybuy@mail.ru

²University of Economics and Law «CROC», 03113, Kyiv, Lagernay st., 30-32,
e-mail: krok@krok.edu.ua

³Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova st., 4d,
e-mail: slm-klm@ukr.net

Статтю представив чл.-кор. НАН України, проф. Анісімов А.В.

Теоретико-множинні конструкції успішно застосовуються в багатьох прикладних областях, зокрема в моделях даних [1] та в табличних алгебрах [2].

Ефективність застосування теорії бінарних відношень в табличних алгебрах пов'язана з існуванням простих та природніх зображень сигнатурних операцій за допомогою конструкцій проекції та відношення сумісності.

Нижче наводяться результати для вказаних теоретико-множинних конструкцій.

Зафіксуємо універсум D , елементи якого позначимо $x, y, z, \dots$. Підмножини універсуму позначимо $X, Y, \dots$, бінарні відношення на D – $U, V, \dots$; функціональні бінарні відношення – $f, g, \dots$. Область означеності (рос. – „область определенности” [3, гл. 1, § 1, с. 20]) функції f позначимо $domf$, вона співпадає з проекцією функції (як бінарного відношення)

за першою компонентою – $domf = \pi_1^2 f$. Тут i далі, $\pi_i^2 U$ – проекція бінарного відношення за i -ю компонентою, $i = 1, 2$.

Як звичайно, повним образом множини X відносно відношення U (за іншою термінологією образом множини відносно відношення) назвемо множину $U[X] = \{y \mid \exists x(x \in X \ \& \ \langle x, y \rangle \in U)\}$.

Виходячи з наступного очевидного зображення проекції відношення (за першою чи другою компонентою) через повний образ відношення відносно селектора

$$\pi_i^2 U = I_i^2[U], \quad i = 1, 2; \quad (1)$$

де $I_i^2 : D \times D \rightarrow D$, $I_1^2(x, y) = x$, $I_2^2(x, y) = y$ – селектори, вкажемо загальнозначні властивості проекції; крім того наведемо цікаві властивості проекції, що перевіряються безпосередньо:

1) $\pi_i^2 \bigcap_j U_j \subseteq \bigcap_j \pi_i^2 U_j$ (верхня оцінка проекції перетину); $\pi_i^2 \bigcup_j U_j = \bigcup_j \pi_i^2 U_j$ (дистрибутивність проекції відносно об'єднань);

2) $\pi_1^2(X \times Y) = X$, якщо $Y \neq \emptyset$;
 $\pi_2^2(X \times Y) = Y$, якщо $X \neq \emptyset$ (проекції декартового добутку);

3) $\pi_i^2 U = \emptyset \Leftrightarrow U = \varepsilon$, $i = 1, 2$ (критерій порожності проекції);

4) $U \subseteq \pi_1^2(U) \times \pi_2^2(U)$ (верхня оцінка відношення в термінах декартового добутку проекцій);

5) $\pi_2^2 U = U[D]$ (вираз проекції за другою компонентою через повний образ);

6) $f \approx g \Rightarrow \text{dom}(f \setminus g) = \text{dom}(f) \setminus \text{dom}(g)$ (достатня умова дистрибутивності проекції за першою компонентою відносно різниці функцій).

7) $U_1 \subseteq U_2 \Rightarrow \pi_i^2 U_1 \subseteq \pi_i^2 U_2$, де $i = 1, 2$ (монотонність проекції за першою та другою компонентою).

В п. 6 використане $\approx$ – бінарне відношення сумісності відношень, зокрема, функцій:
 $U \overset{\text{def}}{\approx} V \Leftrightarrow U \mid X = V \mid X$, тут $X = \pi_1^2 U \cap \pi_1^2 V$ – перетин проекцій вихідних відношень за першою компонентою [2, підрозд. 1.3, с. 25].

Очевидно, що відношення сумісності рефлексивне, симетричне (що впливає з однойменних властивостей рівності), але нетранзитивне. Таким чином відношення сумісності є відношенням толерантності (в розумінні, наприклад, [4, гл. 3, § 1, с. 80]. Також очевидно, що порожнє відношення сумісне з довільним відношенням – $\varepsilon \approx U$, оскільки обмеження зберігає порожню множину [5, твердження 1.11, п. 3].

Властивості 1), 3), 7) впливають з зображення (1) та відповідних властивостей повного образу; властивості 2), 4), 5) перевіряються безпосередньо; далі зупинимось на доведенні властивості 6).

Почнемо з леми про зв'язок між функціональністю та ін'єктивністю.

Лема 1 (зв'язок між функціональністю та ін'єктивністю, критерій функціональності). Відношення U є функціональним $\Leftrightarrow$ обмеження вигляду $I_1^2 \mid U$ є ін'єктивним. $\square$

Доведення. Необхідність встановимо від супротивного: нехай відношення U

функціональне, але обмеження $I_1^2 \mid U$ не є ін'єктивним. Тоді існують різні пари $\langle x, y \rangle, \langle x', y' \rangle \in U$, такі, що $x = x'$. Отже, $y \neq y'$, що суперечить функціональності відношення U . $\square$

Достатність доводиться аналогічно. $\square$

Наслідок 1. (достатня умова дистрибутивності проекції за першою компонентою відносно різниці функцій) Виконується імплікація

$$f \approx g \Rightarrow \text{dom}(f \setminus g) = \text{dom}(f) \setminus \text{dom}(g). \square$$

Доведення. Нехай $f \approx g$, тоді згідно з [2, підрозд. 2.7, лема 2.7.1, с. 49] відношення $f \cup g$ функціональне. Значить, з огляду на попередню лему 1 обмеження $I_1^2 \mid (f \cup g)$ є ін'єктивним.

Залишається застосувати достатню умову дистрибутивності повного образу відносно різниці (5, твердження 1.3, п. 3) та зображення (1):

$$\begin{aligned} \text{dom}(f \setminus g) &= I_1^2[f \setminus g] = I_1^2[f] \setminus I_1^2[g] = \\ &= \text{dom}(f) \setminus \text{dom}(g) \end{aligned} \quad \square$$

Далі буде показано, що імплікація останнього наслідку обертається. Почнемо з узагальнення доведеного наслідку 1 для відношень.

Твердження 1 (критерій дистрибутивності проекції за першою компонентою відносно різниці відношень). Виконується еквівалентність $U \mid X \subseteq V \Leftrightarrow \pi_1^2(U \setminus V) = \pi_1^2 U \setminus \pi_1^2 V$, де

$$X = \pi_1^2 U \cap \pi_1^2 V. \quad \square$$

Доведення почнемо з необхідності. Отже, нехай $U \mid X \subseteq V$, встановимо рівність $\pi_1^2(U \setminus V) = \pi_1^2 U \setminus \pi_1^2 V$, тобто, враховуючи зображення (1), – рівність $I_1^2[U \setminus V] = I_1^2[U] \setminus I_1^2[V]$.

Скористатися достатньою умовою дистрибутивності повного образу відносно різниці, як було зроблено в доведенні попереднього наслідку 1, не можна, оскільки, згідно з лемою 1, для цього потрібна функціональність відношення $U \cup V$, що не виконується в загальному випадку. Тому проведемо пряме доведення.

Згідно з нижньою оцінкою повного образу різниці [5, п. 7 твердження 1.2] виконується включення $I_1^2[U \setminus V] \supseteq I_1^2[U] \setminus I_1^2[V]$; таким чином, залишається встановити обернене включення $I_1^2[U \setminus V] \subseteq I_1^2[U] \setminus I_1^2[V]$.

Отже нехай $x \in I_1^2[U \setminus V]$, тоді існує елемент y , такий, що $\langle x, y \rangle \in U, \langle x, y \rangle \notin V$. Звідси випливає, що $x \in I_1^2[U]$ і залишається встановити, що $x \notin I_1^2[V]$. Зробимо це від супротивного: нехай $x \in I_1^2[V]$, тоді $x \in I_1^2[U] \cap I_1^2[V] = \pi_1^2 U \cap \pi_1^2 V = X$; значить, маємо $\langle x, y \rangle \in U | X \subseteq V$, звідки випливає, що $\langle x, y \rangle \in V$ – прийшли до суперечності. $\square$

Достатність встановимо також від супротивного: нехай $\pi_1^2(U \setminus V) = \pi_1^2 U \setminus \pi_1^2 V$, але включення $U | X \subseteq V$ не виконується. Звідси випливає існування елементів x, y , таких, що $x \in X = \pi_1^2 U \cap \pi_1^2 V$, $\langle x, y \rangle \in U$, але $\langle x, y \rangle \notin V$. Значить, $\langle x, y \rangle \in U \setminus V$, тобто $x \in \pi_1^2(U \setminus V) = \pi_1^2 U \setminus \pi_1^2 V$; таким чином, $x \notin \pi_1^2 V$ – прийшли до суперечності. $\square$

Наслідок 2 (характеристична властивість відношення сумісності, достатня умова дистрибутивності проекції за першою компонентою відносно різниці).

Виконується еквівалентність $U \approx V \Leftrightarrow \pi_1^2(U \setminus V) = \pi_1^2 U \setminus \pi_1^2 V \ \& \ \pi_1^2(V \setminus U) = \pi_1^2 V \setminus \pi_1^2 U$. $\square$

Доведення. Як і раніше позначимо $X = \pi_1^2 U \cap \pi_1^2 V$. З властивостей обмеження випливає наступна проста еквівалентність

$$U | X \subseteq V \Leftrightarrow U | X \subseteq V | X. \quad (2)$$

Дійсно, з спадності обмеження [5, твердження 1.11, п. 6] тривіально випливає імплікація $U | X \subseteq V | X \Rightarrow U | X \subseteq V$, а обернена імплікація випливає з монотонності та ідемпотентності обмеження [5, твердження 1.11, пп. 1, 5]; до обох частин включення $U | X \subseteq V$ треба просто застосувати обмеження за множиною X .

Маючи еквівалентність (2), встановимо необхідність. Нехай $U \approx V$, тоді за означенням відношення сумісності виконується рівність $U | X = V | X$, тобто $U | X \subseteq V | X$ та навпаки $U | X \supseteq V | X$. Залишається застосувати еквівалентність (2) та твердження 1 (критерій дистрибутивності проекції за першою компонентою відносно різниці). $\square$

Встановимо достатність. Нехай $\pi_1^2(U \setminus V) = \pi_1^2 U \setminus \pi_1^2 V$ та $\pi_1^2(V \setminus U) = \pi_1^2 V \setminus \pi_1^2 U$. З твердження 1 випливає, що $U | X \subseteq V$ та $V | X \subseteq U$; отже, враховуючи

еквівалентність (2), маємо $U | X \subseteq V | X$ та $V | X \subseteq U | X$, тобто $U | X = V | X$; залишається просто скористатися означенням відношення сумісності. $\square$

З останнього наслідку випливає, зокрема, наслідок 1 та його наступне узагальнення, що полягає у встановленні оберненої імплікації.

Наслідок 1'. (характеристична властивість відношення сумісності функцій). Виконується еквівалентність

$$f \approx g \Leftrightarrow \text{dom}(f \setminus g) = \text{dom}(f) \setminus \text{dom}(g). \quad \square$$

1) Доведення. Виходячи з наслідку 1 та симетричності відношення сумісності досить довести достатність. Отже, нехай $\text{dom}(f \setminus g) = \text{dom}(f) \setminus \text{dom}(g)$; з твердження 1 випливає, що $f | X \subseteq g$, де $X = \overset{\text{def}}{\text{dom} f \cap \text{dom} g}$; звідси, з огляду на еквівалентність (2) з доведення наслідку 2, маємо включення $f | X \subseteq g | X$; залишається застосувати наступну імплікацію

$$f \subseteq g \ \& \ X \subseteq \text{dom} f \Rightarrow f | X = g | X \quad (3)$$

[5, п. 10 твердження 1.11] та врахувати, що функції останнього включення мають однакову область означеності (а саме X): отже $f | X = g | X$, тобто $f \approx g$. $\square$

Як бачимо, характеристична властивість сумісності функцій (наслідок 1') виглядає більш просто ніж для відношень загального виду (наслідок 2); справа в тому, що саме для функцій виконується імплікація (3). Зауважимо також, що прості приклади та врахування твердження 1 (критерій дистрибутивності проекції за першою компонентою відносно різниці відношень) показують: спростити наслідок 2, вилучаючи один з членів кон'юнкції в правій частині, не можна.

Наступне твердження 2 є аналогом твердження 1 та наслідку 2 для проекції (за першою компонентою) перетину відношень (точніше кажучи, аналогом для дистрибутивності проекції за першою компонентою відносно перетинів). В доведенні будуть потрібні дві наступні леми (причому лема 2 використовується для доведення леми 3). Далі всюди для

спрощення запису повні образи синглтонів $\{x\}$ будемо позначати через $U[x]$.

Лема 2. Виконуються такі твердження:

- 1) $U = \bigcup_{x \in \pi_1^2 U} \{x\} \times U[x]$;
- 2) $U \neq V \ \& \ \pi_1^2 U = \pi_1^2 V \Rightarrow \Rightarrow \exists x (x \in \pi_1^2 U \ \& \ U[x] \neq V[x])$. $\square$

Доведення. Перший пункт перевіряється безпосередньо, а другий випливає з першого (формальне доведення проводиться від супротивного)¹. $\square$

Другий пункт доведеної леми має таку змістовну інтерпретацію: якщо різні відношення мають однакові проєкції за першою компонентою, то існує елемент цієї проєкції, на якому відношення „розрізняються”, цілком зрозуміло також, що імплікація другого пункту не обертається.

Лема 2.3 (критерій сумісності відношень). Виконується еквівалентність $U \approx V \Leftrightarrow U \approx U \setminus V \ \& \ U \approx V \setminus U$. $\square$

Доведення. Почнемо з необхідності. Нехай $U \approx V$, спочатку покажемо, що $U \approx U \setminus V$. З наслідку 2 випливає рівність $\pi_1^2(U \setminus V) = \pi_1^2 U \setminus \pi_1^2 V$; таким чином, $\pi_1^2 U \cap \pi_1^2(U \setminus V) = \pi_1^2 U \setminus \pi_1^2 V$. Отже, позначаючи

$X = \pi_1^2 U \setminus \pi_1^2 V$, треба перевірити рівність $U|X = (U \setminus V)|X$. Дійсно, використовуючи дистрибутивність обмеження відносно об'єднань [5, твердження 1.11, п. 8)], маємо

$$U|X = (U \setminus V \cup U \cap V)|X = \\ = (U \setminus V)|X \cup (U \cap V)|X$$

Тепер, виходячи з критерію порожності обмеження [5, твердження 1.11, п. 3)] та загальних властивостей проєкції, покажемо, що другий доданок правої частини останньої рівності насправді порожній; дійсно:

$$\pi_1^2(U \cap V) \cap X \subseteq \pi_1^2 U \cap \pi_1^2 V \cap X = \\ = \pi_1^2 U \cap \pi_1^2 V \cap (\pi_1^2 U \setminus \pi_1^2 V) = \emptyset$$

Шукана рівність $U|X = (U \setminus V)|X$ встановлена, а з нею і твердження $U \approx U \setminus V$. $\square$

¹ Зауважимо, що рівність першого пункту виконується і для порожнього відношення U (з використанням стандартної домовленості про об'єднання за порожньою множиною індексів $\bigcup_{i \in I} A_i = \emptyset$, якщо $I = \emptyset$).

Друге твердження $U \approx V \setminus U$ встановлюється аналогічно (треба скористатися тим фактом, що згідно з наслідком 2 проєкції за першою компонентою відношень U та $V \setminus U$ взагалі не перетинаються за умови $U \approx V$). $\square$

Доведемо достатність від супротивного. Нехай $U \approx U \setminus V$ та $U \approx V \setminus U$, але відношення U , V не сумісні. Тоді, $U|X \neq V|X$, де

$X = \pi_1^2 U \cap \pi_1^2 V$. Згідно з п. 2 леми 2 існує елемент x_0 , такий, що $x_0 \in X$, причому $U[x_0] \neq V[x_0]$. Оскільки множини $U[x_0], V[x_0]$ не рівні, то можливі три наступних випадки. В перших двох випадках ці множини порівнювані (відносно теоретико-множинного включення), в третьому – непорівнювані.

1. $U[x_0] \subset V[x_0]$. Оскільки множина $U[x_0]$ непорожня, то існують елементи y_0, z_0 , такі, що $y_0 \in U[x_0]$, $z_0 \in V[x_0] \setminus U[x_0]$ (див. рис. 1).

Покладемо $X' = \pi_1^2 U \cap \pi_1^2 (V \setminus U)$. З належностей $\langle x_0, y_0 \rangle \in U$, $\langle x_0, z_0 \rangle \in V \setminus U$ випливає, що $x_0 \in X'$.

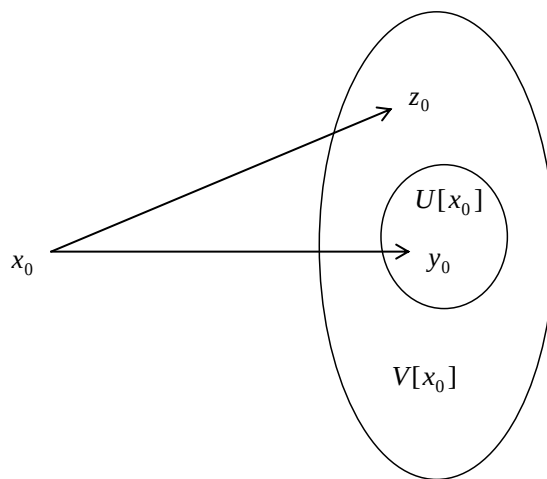


Рис. 1. До п. 1 доведення достатності леми 3

Ідея подальшого доведення полягає у тому, що відношення $U, V \setminus U$, всупереч їхній сумісності, „ведуть себе по-різному” на елементі x_0 .

Тепер використовуючи, що $U \approx V \setminus U$, прийдемо до суперечності в першому випадку. Дійсно, $\langle x_0, y_0 \rangle \in U$, де $x_0 \in X'$, тобто $\langle x_0, y_0 \rangle \in U|X'$. З сумісності $U \approx V \setminus U$ та означення множини X' випливає належність $\langle x_0, y_0 \rangle \in (V \setminus U)|X'$. Тим більше

$\langle x_0, y_0 \rangle \in V \setminus U$, звідки $\langle x_0, y_0 \rangle \notin U$ – суперечність. $\square$

2. $V[x_0] \subset U[x_0]$ (відношення U і V міняються ролями). Оскільки множина $V[x_0]$ непорожня, то існують елементи y_0, z_0 , такі, що $y_0 \in V[x_0]$, $z_0 \in U[x_0] \setminus V[x_0]$ (див. рис. 2).
Покладемо $X'' = \pi_1^2 U \cap \pi_1^2 (U \setminus V)$; оскільки проекція монотонна, то очевидно, що $X'' = \pi_1^2 (U \setminus V)$. З належності $\langle x_0, z_0 \rangle \in U \setminus V$, випливає, що $x_0 \in X''$.

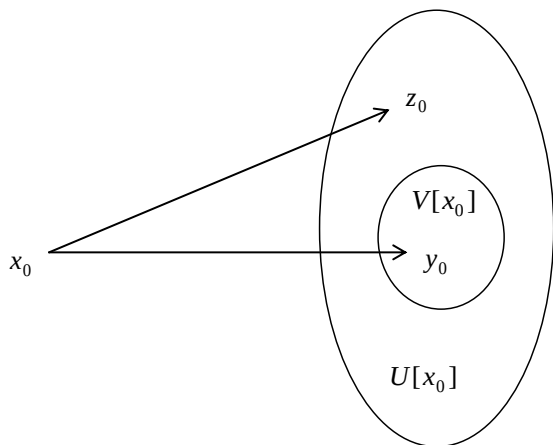


Рис. 2. До п. 2 доведення достатності леми 3

Ідея подальшого доведення полягає у тому, що відношення $U, U \setminus V$, всупереч їх сумісності, „ведуть себе по-різному” на елементі x_0 .

Використовуючи, що $U \approx U \setminus V$, прийдемо до протиріччя в другому випадку, що розглядається. Дійсно, $\langle x_0, y_0 \rangle \in U$, де $x_0 \in X''$, тобто $\langle x_0, y_0 \rangle \in U \mid X''$. З сумісності $U \approx U \setminus V$ та означення множини X'' випливає належність $\langle x_0, y_0 \rangle \in (U \setminus V) \mid X''$. Тим більше $\langle x_0, y_0 \rangle \in U \setminus V$, звідки $\langle x_0, y_0 \rangle \notin U$ – протиріччя. $\square$

3. $U[x_0] \setminus V[x_0] \neq \emptyset$ і $V[x_0] \setminus U[x_0] \neq \emptyset$. Тоді існують елементи y_0, z_0 , такі, що $y_0 \in U[x_0] \setminus V[x_0]$, $z_0 \in V[x_0] \setminus U[x_0]$ (див. рис. 3).
Покладемо $X''' = \pi_1^2 U \cap \pi_1^2 (V \setminus U)$.

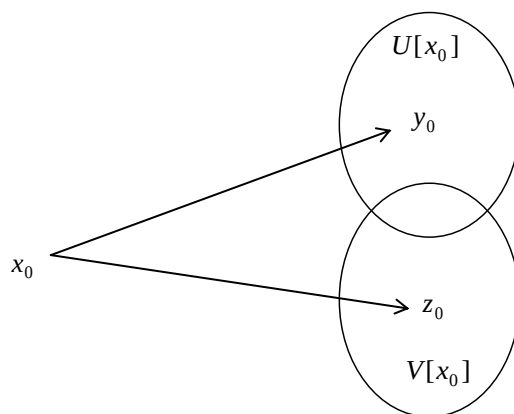


Рис. 3. До п. 3 доведення достатності леми 3

Оскільки $\langle x_0, y_0 \rangle \in U \setminus V$, то $x_0 \in \pi_1^2 (U \setminus V) \subseteq \pi_1^2 U$; аналогічно, оскільки $\langle x_0, z_0 \rangle \in V \setminus U$, то $x_0 \in \pi_1^2 (V \setminus U)$. Отже $x_0 \in X'''$.

Ідея подальшого доведення полягає у тому, що відношення $U, V \setminus U$, всупереч їхньої сумісності, „ведуть себе по-різному” на елементі x_0 .

Використовуючи умову $U \approx V \setminus U$, прийдемо до протиріччя в третьому випадку, що розглядається. Дійсно, $\langle x_0, y_0 \rangle \in U$, де $x_0 \in X'''$, тобто $\langle x_0, y_0 \rangle \in U \mid X'''$. З сумісності $U \approx V \setminus U$ та означення множини X''' випливає належність $\langle x_0, y_0 \rangle \in (V \setminus U) \mid X'''$. Тим більше $\langle x_0, y_0 \rangle \in V \setminus U$, звідки $\langle x_0, y_0 \rangle \notin U$ – суперечність. $\square$

Твердження 2 (достатня умова дистрибутивності проекції відносно перетину відношень, характеристична ознака відношення сумісності функцій, критерій дистрибутивності проекції відносно перетину функцій). Для відношень виконується імплікація $U \approx V \Rightarrow \pi_1^2 (U \cap V) = \pi_1^2 U \cap \pi_1^2 V$, яка в загальному випадку не обертається. Для функцій виконується еквівалентність $f \approx g \Leftrightarrow \text{dom}(f \cap g) = \text{dom} f \cap \text{dom} g$. $\square$

Доведення. Почнемо з імплікації для відношень, нехай відношення сумісні – $U \approx V$. З леми 3 випливає, що тоді і $U \approx U \setminus V$. Використовуючи наслідок 2 та очевидні теоретико-множинні співвідношення, маємо ланцюжок рівностей:

² Отже, перша імплікація обертається тільки для функцій.

$$\pi_1^2(U \cap V) = \pi_1^2(U \setminus (U \setminus V)) = \pi_1^2 U \setminus \pi_1^2(U \setminus V) = \\ = \pi_1^2 U \setminus (\pi_1^2 U \setminus \pi_1^2 V) = \pi_1^2 U \cap \pi_1^2 V.$$

Тепер наведемо приклад, коли встановлена імплікація не обертається. Нехай x, y, z, t – попарно різні елементи, покладемо $U = \{ \langle x, y \rangle, \langle x, t \rangle \}$ та $V = \{ \langle x, z \rangle, \langle x, t \rangle \}$ (рис. 4). Очевидно, що ці відношення несумісні, але $\pi_1^2(U \cap V) = \pi_1^2 U \cap \pi_1^2 V = \pi_1^2 U = \pi_1^2 V = \{x\}$.

Нарешті покажемо, що для функцій імплікація обертається, тобто $\text{dom}(f \cap g) = \text{dom} f \cap \text{dom} g \Rightarrow f \approx g$. Очевидно, що $f \cap g$ – функція, причому $f \cap g \subseteq f$ та $f \cap g \subseteq g$. З наступної імплікації $f \subseteq g \Rightarrow f = g \mid \text{dom} f$, [5, п. 10 твердження 1.11] випливає, що $f \cap g = f \mid \text{dom}(f \cap g) = g \mid \text{dom}(f \cap g)$. Але $\text{dom}(f \cap g) = \text{dom} f \cap \text{dom} g$, отже, $f \mid (\text{dom}(f) \cap \text{dom}(g)) = g \mid (\text{dom}(f) \cap \text{dom}(g))$, тобто $f \approx g$. □

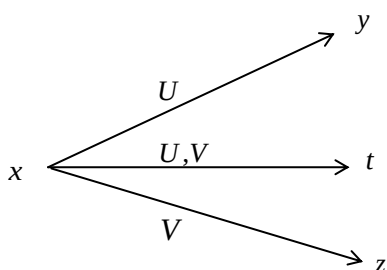


Рис. 4. Приклад несумісних відношень U, V , таких, що $\pi_1^2(U \cap V) = \pi_1^2 U \cap \pi_1^2 V$

Висновки

В роботі розглянуто теоретико-множинні конструкції, а саме проекцію та відношення сумісності. Встановлено критерій

дистрибутивності проекції за першою компонентою відносно різниці відношень, характеристичну властивість відношення сумісності, достатню умову дистрибутивності проекції за першою компонентою відносно різниці, характеристичну властивість відношення сумісності функцій, достатню умову дистрибутивності проекції відносно перетину відношень, характеристичну ознаку відношення сумісності функцій, критерій дистрибутивності проекції відносно перетину функцій.

Список використаних джерел

1. Buy D. Formalization of model entity-relationship / D. Buy, L. Silvestru // Kyiv: Publishing center "Kyiv University", 2011. – 176 p. (in Ukrainian).
2. Relational databases : tabular algebra and SQL-like language / VN Red'ko, Yu.Y. Bronte, DB Bui, SA Polyakov. - Kyiv : Publishing House "Akademperiodika" 2001. - 198 p. (in Ukrainian).
3. A.I. Maltsev Algorithms and functions rekursivnye / A.I. Maltsev. – Moscow : Nauka, 1965. - 391 p. (in Russian).
4. J.A. Schroeder Equality, similarity, order / YA Schrader. - Moscow : Nauka, 1971. – 255 p. (in Russian).
5. Kahuta N.D Applications of set-theoretic constructions of whole image, restrictions, finality and consistency in tabular databases: Dissertation of candidate of physical and mathematical sciences : 01.05.03 – Mathematical and software of computers and systems / Kahuta Nadezhda Dmitrievna. - Kyiv, 2010. – 116 p. (in Ukrainian).

Надійшла до редколегії 17.06.13

УДК 004.655

Буй Д.Б.¹, д. ф.-м. н.,
Поляков С.А.², к. ф.-м. н.,
Гришко Ю.А.³, к. ф.-м. н.

Формализация модели данных, используемой в NoSQL базах данных

В статье строятся модели данных, лежащие в основе NoSQL баз данных. Построение ведется на основе множеств и мультимножеств. Вводятся отношения «быть поддокументом» и «быть подзаписью». Доказывается, что эти отношения являются отношениями предпорядка.

Ключевые слова: модель данных, NoSQL, отношение порядка, отношение предпорядка.

E-mail:

¹buy@unicyb.kiev.ua,

²polyakov@unicyb.kiev.ua,

³iuliia.hryshko@gmail.com

Buy D.B.¹
Polyakov S.A.²
Hryshko I.A.³

Formal specification of the NoSQL data model

The data models which form the basis of the NoSQL DBMS are built. The sets and bugs are used to define the models. The special relations called subdocument and subrecord are introduced. It is proven those relations are preorder.

Key words: data model, NoSQL, order, preorder.

E-mail:

¹buy@unicyb.kiev.ua,

²polyakov@unicyb.kiev.ua,

³iuliia.hryshko@gmail.com

Статью представил д. ф.-м. н., проф. Анісімов А.В.

Введение

Существующие NoSQL СУБД базируются на нескольких моделях данных. Можно говорить скорее о новой идеологии построения СУБД, альтернативной реляционной, чем о единой платформе, лежащей в основе NoSQL баз данных [1]. Одним из наиболее распространенных типов NoSQL СУБД являются документно-ориентированные системы, подобные MongoDB [2] и CouchDB, и основанные на формате представления документов JSON. Ниже будут рассмотрены формальные модели, описывающие структуры данных, используемые документно-ориентированных NoSQL СУБД, и изучены некоторые свойства данных моделей.

Построение ведется на основе композиционного подхода к программированию [3]. В свое время он был успешно применен при описании семантики реляционных баз данных и языка SQL [4]. Построенные модели использует в своей основе множества, мультимножества и именные множества [5].

Отношения включения документов на множествах

Множество всех конечных подмножеств X будем обозначать как $2_{fin}^X = \{X' | X' \subseteq X \text{ \& } X' - \text{конечно}\}$.

Пусть дано множество атомарных данных D и множество имен V . Именные множества, которые обозначим как D^V , определяются как конечные отображения из множества имен V во множество данных D , т. е. $D^V = \{A | A: V \rightarrow D, V' \in 2_{fin}^V\}$.

Множество записей RC , и документов DC , строится индуктивно. А именно, множество записей ранга 0 совпадает с D^V . Обозначим это множество через RC^0 . Множество документов ранга 0, обозначаемое как DC^0 , есть множество всех конечных подмножеств RC^0 , т.е. $DC^0 = 2_{fin}^{RC^0}$. Пусть заданы записи и документы ранга i . Тогда записи ранга $i+1$ задаются как $RC^{i+1} = (D \cup_{j=0}^i DC^j)^V$. Т.е. значением имени может быть либо атомарное данное, либо документ одного из предыдущих рангов. Соответственно документ ранга $i+1$ задается как множество всех конечных подмножеств записей ранга $i+1$, т.е. $DC^{i+1} = 2_{fin}^{RC^{i+1}}$.

Так как $DC^0 \subset DC^1 \subset DC^2 \subset \dots$ монотонно возрастающая последовательность по построению, то она имеет предел $DC = \lim_{i \rightarrow \infty} DC^i = \bigcup_{i=0}^{\infty} DC^i$. Аналогично для записей $RC = \lim_{i \rightarrow \infty} RC^i = \bigcup_{i=0}^{\infty} RC^i$.

С учетом монотонности определение записей $i+1$ -го ранга можно переписать как $RC^{i+1} = (D \cup DC^i)^V$.

По аналогии с отношением включения $\subseteq$ для множеств, для документов и записей введем отношения быть поддокументом и быть подзаписью, которые учитывают внутреннюю структуру документов и записей. Обозначим эти отношения $sdoc$ (subdocument) и $srec$ (subrecord) соответственно. Отношения вводятся индуктивно. Пусть $r_1, r_2 \in RC^0$. Тогда $r_1 srec r_2$ тогда и только тогда, когда $r_1 \subseteq r_2$. Аналогично, для документов нулевого ранга, $d_1 sdoc d_2$ тогда и только тогда, когда $\forall r_1 \in d_1 \exists r_2 \in d_2 (r_1 \subseteq r_2)$.

Пусть эти отношения определены для документов и записей i -го ранга. Тогда, для записей $i+1$ -го ранга отношение $r_1^{i+1} srec r_2^{i+1}$, означает, что множество имен записи r_1^{i+1} включается во множество имен записи r_2^{i+1} . А значения при одинаковых именах одновременно либо являются атомарными данными и совпадают между собой, либо являются документами. Причем, тогда их ранг меньше $i+1$, и документ из r_1^{i+1} является поддокументом соответствующего документа из r_2^{i+1} .

d_1^{i+1} является поддокументом d_2^{i+1} тогда и только тогда, когда для любой записи $r_1 \in d_1^{i+1}$, существует запись $r_2 \in d_2^{i+1}$, для которых $r_1 srec r_2$. Отметим, что для r_1 , вообще говоря, может существовать несколько соответствующих записей в документе d_2^{i+1} .

Введенные таким образом отношения $srec$ и $sdoc$ являются отношениями предпорядка. Т.е. они рефлексивны, транзитивны, но не антисимметричны.

Докажем рефлексивность. Доказательство проводится индукцией по рангу записей и документов.

База индукции. Для записей и документов нулевого ранга утверждение очевидно.

Шаг индукции. Пусть утверждение выполняется для записей и документов i -го ранга. Тогда покажем, что для записей $i+1$ -го ранга рефлексивность также имеет место, т.е. $r^{i+1} srec r^{i+1}$.

Действительно:

- множество имен включается в само себя;
- если значение имени является атомарное данное, то оно совпадает само с собой;
- если значением имени является документ, то, по построению, его ранг меньше $i+1$, следовательно, по предположению индукции, он является поддокументом самого себя.

Для документов $i+1$ -го ранга рефлексивность также имеет место, так как каждая запись $i+1$ -го ранга, как показано выше, является подзаписью самой себя.

Докажем теперь транзитивность. А именно, что имеют место формулы $r_1 srec r_2 \& r_2 srec r_3 \Rightarrow r_1 srec r_3$ и $d_1 sdoc d_2 \& d_2 sdoc d_3 \Rightarrow d_1 sdoc d_3$. Доказательство будем проводить индукцией по рангу документов и записей.

База индукции. Для документов и записей нулевого ранга транзитивность имеет место, так как отношения быть поддокументом и быть подзаписью для документов и записей, в этом случае, совпадают с отношением быть подмножеством для множеств.

Шаг индукции. Пусть транзитивность выполняется для документов и записей i -го ранга. Покажем, что тогда она будет иметь место для документов и записей $i+1$ -го ранга.

Сначала рассмотрим записи. Пусть $r_1^{i+1} srec r_2^{i+1}$ и $r_2^{i+1} srec r_3^{i+1}$.

Так как, по определению, множество имен записи r_1^{i+1} включается во множество имен записи r_2^{i+1} , и множество имен записи r_2^{i+1} включается во множество имен записи r_3^{i+1} , то множество имен записи r_1^{i+1} включается во множество имен записи r_3^{i+1} .

Пусть значением имени n в записи r_1^{i+1} является атомарное данное. Это означает, по определению, что имя n принадлежит множеству имен записи r_2^{i+1} и его значением в этой записи выступает то же самое атомарное данное. Но тогда это имя должно принадлежать множеству имен записи r_3^{i+1} и иметь в ней такое же значение. Следовательно, если некоторое имя в записи r_1^{i+1} имеет атомарное значение, то оно имеет такое же значение в записи r_3^{i+1} .

Пусть значением имени n в записи r_1^{i+1} является некий документ d_1 . Тогда значением имени n в записи r_2^{i+1} является, вообще говоря, другой документ d_2 , и значением имени n в записи r_3^{i+1} также является документ d_3 , причем имеет место $d_1 sdoc d_2$ и $d_2 sdoc d_3$. По построению, ранг документов d_1, d_2, d_3 строго меньше $i+1$. Следовательно, по предположению индукции, $d_1 sdoc d_3$.

Теперь перейдем к документам $i+1$ -го ранга.

Пусть $d_1^{i+1} sdoc d_2^{i+1}$ и $d_2^{i+1} sdoc d_3^{i+1}$. По определению, имеем $\forall r_1 \in d_1^{i+1} \exists r_2 \in d_2^{i+1} (r_1 srec r_2)$ и $\forall r_2 \in d_2^{i+1} \exists r_3 \in d_3^{i+1} (r_2 srec r_3)$. Отсюда

получаем $\forall r_1 \in d_1^{i+1} \exists r_2 \in d_2^{i+1} \exists r_3 \in d_3^{i+1}$
($r_1 \text{ srec } r_2 \& r_2 \text{ srec } r_3$). Следовательно, $r_1 \text{ srec } r_3$.

В то же время антисимметричность для этих отношений не имеет места. Действительно, рассмотрим следующий пример. Пусть $d_1 = \{(a, 1), (b, 2)\}, \{(a, 1)\}$, а $d_2 = \{(a, 1), (b, 2)\}$. Тогда имеем $d_1 \text{ sdoc } d_2$ и $d_2 \text{ sdoc } d_1$, но очевидно, что $d_1 \neq d_2$.

Отношения включения документов на мультимножествах

В реальных документно-ориентированных СУБД записи могут дублироваться. Ситуация аналогичная таблицам в реляционных СУБД, где допускается дублирование строк таблицы. Поэтому дальнейшее уточнение построенной модели данных необходимо проводить с использованием мультимножеств. А именно, рассмотрим возможность повторения записей в пределах документа.

Сначала введем некоторые определения теории мультимножеств, необходимые для построения данной модели [5].

Мультимножество α с основой U – это функция $\alpha: U \rightarrow N^+$, где U – некоторое множество, а N^+ – множество натуральных чисел без нуля. Запись M_U будет использоваться для обозначения всех конечных мультимножеств с основой U .

Теперь определим множества записей RC_M и документов DC_M для новой модели. Определения будут строиться индуктивно. Множество записей ранга 0 совпадает с совокупностью именных множеств, т.е. $RC_M^0 = D^V$. Множество документов ранга 0 определяется как совокупность всех конечных мультимножеств, основами которых выступают конечные множества записей ранга 0, т.е. $DC_M^0 = \bigcup_{d \in 2_{fin}^{RC_M^0}} M_d$. Пусть заданы записи и документы ранга i . Тогда записи ранга $i+1$ задаются, как и предыдущем случае $RC_M^{i+1} = (D \cup_{j=0}^i DC_M^j)^V$. Значением имени может быть либо атомарное данное, либо документ одного из предыдущих рангов. Соответственно, документы ранга $i+1$ задаются как конечные мультимножества, основами которых выступают конечные множества записей ранга $i+1$, т.е. $DC_M^{i+1} = \bigcup_{d \in 2_{fin}^{RC_M^{i+1}}} M_d$.

Теперь переопределим отношения быть поддокументом (обозначим его как sdoc_M) и быть подзаписью (обозначим его как srec_M). Отношения вводятся индуктивно.

Для записей нулевого ранга определение не меняется, т.е. $r_1 \text{ srec}_M r_2$ тогда и только тогда, когда $r_1 \subseteq r_2$. Для документов нулевого ранга $d_1 \text{ sdoc}_M d_2$ тогда и только тогда, когда $\forall r_1 \in U_{d_1} \exists r_2 \in U_{d_2} (r_1 \subseteq r_2)$, где U_{d_1} и U_{d_2} основы документов d_1 и d_2 соответственно.

Пусть эти отношения определены для документов и записей i -го ранга. Тогда, для записей $i+1$ -го ранга отношение $r_1^{i+1} \text{ srec}_M r_2^{i+1}$, определяется, так же как и выше, за исключением того, что если значениями совпадающих имен являются документы, то они находятся в отношении sdoc_M , а не sdoc .

Для документов $i+1$ -го ранга d_1^{i+1} является поддокументом d_2^{i+1} тогда и только тогда, когда для любой записи $r_1 \in U_{d_1^{i+1}}$, существует запись $r_2 \in U_{d_2^{i+1}}$, для которой $r_1 \text{ srec}_M r_2$. При таком определении, количество экземпляров записей не учитываются.

Введенные таким образом отношения $\text{sdoc}_M, \text{srec}_M$ также являются отношениями предпорядка. Доказательство проводится аналогично доказательству для случая множеств.

Примеры

Рассмотрим документ, содержащий список книг книжного интернет магазина, сгруппированный по тематике и подтематике. В целях упрощения записи, пару (атрибут, значение) будем записывать в виде атрибут: значение.

```
{ {Тема: Компьютеры,
  { {Подтема: Базы данных,
    Книги: <книги по БД> },
    {Подтема: Интернет,
      Книги: <книги по интернету> }
    } },
  где
  <книги по БД> = {
    {Название: MySQL на примерах,
      Издательство: БХВ-Петербург,
      Авторы: { {ФИО: М. Кузнецов},
                 {ФИО: И. Симдянов} },
      },
    {Название: Технологии анализа данных
      Издательство: БХВ-Петербург
      Авторы: { {ФИО: А. Барсегян},
                 {ФИО: М. Куприянов},
                 {ФИО: В. Степаненко},
                 {ФИО: И. Холод} },
      }
    }
  }
```

<книги по интернету> = {
 { *Название*: HTML5. Недостающее
руководство,
 Издательство: БХВ-Петербург
 Авторы: {{ *ФИО* : Мэтью Мак-Дональд }}
 }}

Модифицируем документ. А именно, добавим
подтему по языкам программирования, а к
информации по книгам добавим год издания.
Документ примет следующий вид:

{{ *Тема*: Компьютеры,
 {{ *Подтема*: Базы данных,
 Книги: <книги по БД> },

 { *Подтема*: Интернет,
 Книги: <книги по интернету> },

 { *Подтема*: Языки программирования,
 Книги: <книги по ЯП> }
 }

}},

где

<книги по БД> = {
 { *Название*: MySQL на примерах,
 Издательство: БХВ-Петербург,
 Авторы: {{ *ФИО*: М. Кузнецов },
 { *ФИО*: И. Симдянов } },
 Год издания: 2009 },
 { *Название*: Технологии анализа данных
 Издательство: БХВ-Петербург
 Авторы: {{ *ФИО* : А. Барсегян },
 { *ФИО* : М. Куприянов },
 { *ФИО* : В. Степаненко },
 { *ФИО* : И. Холод } },
 Год издания: 2007 }
}

<книги по интернету> = {
 { *Название*: HTML5. Недостающее
руководство,

Список литературы:

1. *Sadalage J. Pramod.* NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence / Pramod J. Sadalage, Martin Fowler // Addison-Wesley Professional, 2012. – 192 p.
2. *Chodorow Kristina.* MongoDB: The Definitive Guide / Kristina Chodorow, Michael Dirolf // O'Reilly Media, 2010. – 216 p.
3. *Redko V.N.* Bases of programmology / V.N. Redko // Cybernetics and systems analysis. – 2000. – № 1. – P. 35-57. (in Russian)
4. *Redko V.N.* Relation databases: table algebras and

Издательство: БХВ-Петербург
Авторы: {{ *ФИО* : Мэтью Мак-Дональд }}
Год издания: 2012 }
}}

<книги по ЯП> = {
 { *Название*: Программирование в Clojure
 Издательство: ДМК Пресс
 Авторы: {{ *ФИО* : Чаз Эмерик },
 { *ФИО* : Брайан Карпер },
 { *ФИО* : Кристоф Гранд } }
 Год издания: 2013 }
}}

При такой модификации старый документ
будет являться поддокументом нового.

Выводы

В работе построены две модели данных,
уточняющие структуры данных, используемые в
документно-ориентированных базах данных.

Первая модель основывается на множествах, в
то время, как вторая основывается на
мультимножествах, и допускает дубликаты
записей в пределах одного документа.

Можно провести аналогию с реляционными
базами данных, где таблицы также определяются
как множество записей, либо как
мультимножество записей, что существенно
влияет на определение операций.

Введенные отношения поддокумента и
подзаписи формализуют интуитивное
представление о включении одного документа в
другой. Они расширяют стандартное отношение
подмножества (подмультимножества) на
документы с учетом их внутренней структуры. В
то же время, эти отношения являются
предпорядком, что не позволяет говорить о
полной аналогии с отношением «быть
подмножеством» («быть
подмультимножеством»).

SQL-like languages / V.N. Redko, J.I. Brona, D.B.
Buy, S.A. Polyakov // Kiev: «Akademperiodika»,
2001. – 196 p. (in Ukrainian)

5. *Buy D.B.* About different methods of nominal data definition / D.B. Buy, J.I. Brona // Bulletin of Taras Shevchenko National University. Physics-mathematics science series. – 1994. – P. 186-194. (in Ukrainian)
6. *Buy D.B.* Multisets: definitions, operations, basis properties / D.B. Buy, J.A. Bogatyreva // Proceeding of 5th International conference TAAPSD'2008. – Kiev. – P. 23-26. (in Ukrainian)

Надійшла до редколегії 20.07.2013

УДК 517.977:519.63

Власюк А.П.¹, д.т.н., проф.,
Багнюк О.М.², аспірант

Ідентифікація невідомих параметрів джерел забруднень в двовимірній стаціонарній задачі масопереносу для області з криволінійною межею

Сформульована обернена задача дифузійного перенесення забруднень від точкового джерела, розміщеного в однозв'язній області з криволінійною межею. Знайдено невідомі координати місцеположення джерела на основі інформації про необхідну кількість замірів концентрації в заданих точках. Розв'язок оберненої крайової задачі знайдено з використанням числових конформних відображень.

Ключові слова: джерело забруднення, ідентифікація, скінченне перетворення Фур'є, спряжене рівняння, функція Гріна, числове конформне відображення.

¹ Міжнародний економіко-гуманітарний університет імені акад. Степана Дем'янчука, 33027, Україна, м.Рівне, вул. акад. С. Дем'янчука, 4,
e-mail: A.P.Vlasyuk.rv@gmail.com

² Національний університет водного господарства та природокористування, 33000, м. Рівне, вул. Соборна, 11,
e-mail: olga_bagnyk@ukr.net

A.P. Vlasyuk¹, Doctor of science, prof.,
O.M.Bagnyuk², PhD student

Identification of unknown parameters of pollution sources in two-dimensional stationary mass transfer problem in the domain with curvilinear boundary

Invers problem of diffusion transfer of pollutants from a point source that is located in simply connected domain with curvilinear boundary was formulated. Unknown source location coordinates were found basing on information about the required measurements number of concentration at the given points. Solution of the inverse boundary value problem was found using numerical conformal mappings.

Key words: source of pollution, identification, Fourier finite transformation, adjoint equation, Green function, numerical conformal mapping.

¹ International university of Economics and Humanities named after Academician Stepan Demianchuk, acad. Stepan Demianchuk str., 4, Rivne, 33027, Ukraine
e-mail: A.P.Vlasyuk.rv@gmail.com

² National University of Water Management and Nature Resources Use, Soborna, 11, Rivne, 33000, Ukraine
e-mail: olga_bagnyk@ukr.net

Статтю представив д.т.н., проф. Гаращенко Ф.Г.

1. Вступ. Проблема забруднення навколишнього середовища є важливою задачею сьогодення. Основними джерелами забруднення повітря, води, ґрунту є промислові підприємства (заводи, фабрики, теплові електростанції) і транспорт. Практично всі забруднюючі речовини, що спочатку потрапили в повітря, воду згодом опиняються у ґрунтах. Крім того, ґрунти забруднюються і при надмірному внесенні в них отрутохімікатів та мінеральних добрив, міграції радіонуклідів. Самоочищення ґрунтового середовища відбувається дуже повільно, тому отруйні хімічні речовини

нагромаджуються там. Їх поглинають рослини, вживання яких викликає захворювання людей і тварин. В останній час в екологічній проблематиці різко зріс інтерес до обернених задач, метою яких є ідентифікація тих чи інших параметрів джерела забруднення за даними вимірювання концентрації забруднюючого мігранта.

Перспективним у вирішенні такої проблеми може бути розв'язання задачі ідентифікації джерел забруднення. Її розв'язок дозволяє встановити вклад окремих джерел в забруднення повітря або ґрунту в даній точці. Це необхідно

враховувати при створенні нових джерел – вводу нових промислових підприємств і управлінні викидами існуючих, щоб сумарна їх кількість не перевищувала встановлених норм.

Для визначення можливих джерел забруднення використовуються методи, в основі яких лежить розв'язок обернених задач, метою яких є встановлення тих чи інших параметрів джерела забруднення за даними спостережень.

Вивченню цих задач присвячені роботи таких вчених, як С.І. Ляшка, О.Б. Стелі, О.А. Самарського, Г.І. Марчука, О.М. Білоцерківського, А.Є. Алояна, В.В. Пененка, П.Н. Вабіщевича, Борухова В.Т. та ін.[].

Метою даної роботи є ідентифікація місцеположення джерела забруднення на прикладі двовимірної модельної стаціонарної задачі в криволінійній області.

2. Постановка задачі. Розглянемо задачу чисто дифузійного перенесення забруднення в деякому однорідному середовищі (грунті, повітрі, воді) в криволінійній однозв'язній області G , обмеженої контуром $\tilde{A} = \bigcup_{i=1} \Gamma_i$ від деякого точкового джерела потужності Q , розміщеного в точці $M(x_0, y_0)$ (рис. 1).

Задача полягає в тому, щоб при заданих значеннях параметрів процесу масоперенесення (коефіцієнта дифузії D , потужності джерела забруднення Q та необхідній кількості замірів концентрації в деяких точках $C_i(x_i, y_i)$, $i = \overline{1, N}$ розглядуваної області), потрібно знайти координати (x_0, y_0) місцеположення джерела забруднення в заданій області.

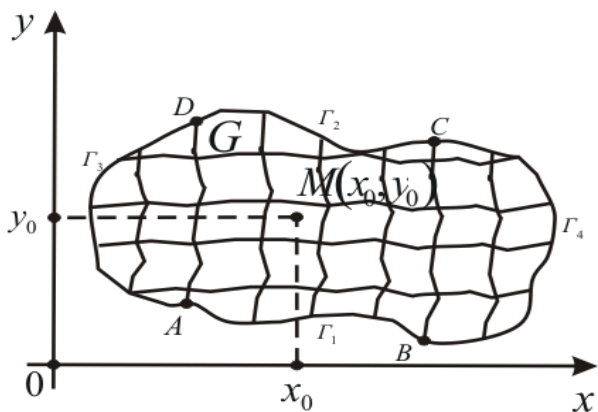


Рис.1. Розміщення джерела забруднення в області з криволінійною межею

3. Математична модель задачі в змінних x, y фізичної області. Математичну модель задачі ідентифікації місцеположення джерела забруднення в заданій області можна описати наступною крайовою задачею:

$$Lc \equiv D \left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} \right) = -Q \delta(x - x_0) \delta(y - y_0), \quad (1)$$

$$\tilde{n}(x, y, x_0, y_0)|_{\tilde{A}} = 0. \quad (2)$$

Тут: $\tilde{n}(x, y, x_0, y_0)$ – концентрація забруднень в точці (x, y) від джерела, розміщеного в точці (x_0, y_0) ; D – коефіцієнт дифузії; Q – потужність точкового джерела забруднень; $\delta(x)$, $\delta(y)$ – дельта-функції Дірака.

Потрібно при заданих вхідних даних: $Q > 0$, $D > 0$ та заміряних значеннях концентрації C_i , $i = \overline{1, N}$ знайти координати (x_0, y_0) точки $\dot{I}$ місцеположення джерела забруднення в заданій області.

4. Числове розв'язання задачі для однозв'язної області з криволінійною межею

4.1. Числове конформне відображення області з криволінійною межею на параметричний прямокутник

Для розв'язання задачі (1), (2), побудуємо числове конформне відображення криволінійної області G на параметричний прямокутник з сторонами a і b (рис.2). Нехай це відображення (пряме і обернене) реалізується відповідно за допомогою пар спряжених гармонічних функцій

$$\begin{cases} \xi = \xi(x, y), \\ \eta = \eta(x, y), \end{cases} \quad \begin{cases} x = x(\xi, \eta), \\ y = y(\xi, \eta). \end{cases}$$

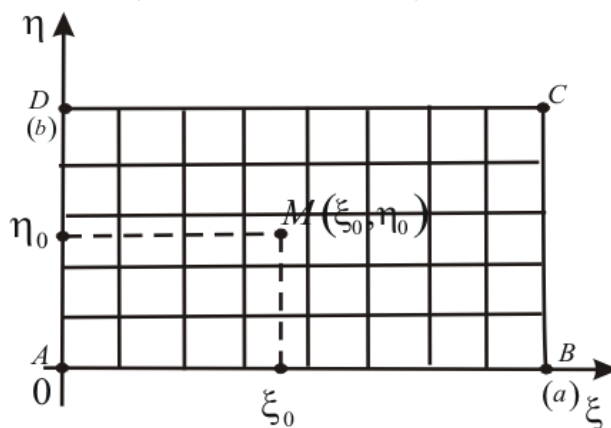


Рис.2. Образ джерела забруднення в параметричному прямокутнику

Відобразимо конформно криволінійну
однозв'язну область на параметричний
прямокутник за допомогою аналітичної функції

$$\zeta(z) = \xi(x, y) + i\eta(x, y),$$

дійсна і уявна частина якої задовольняє умовам
Коші-Рімана:

$$\frac{\partial \xi}{\partial x} = \frac{\partial \eta}{\partial y}, \quad \frac{\partial \xi}{\partial y} = -\frac{\partial \eta}{\partial x}. \quad (3)$$

На межі області G для пари спряжених
гармонічних функцій $\begin{cases} \xi = \xi(x, y) \\ \eta = \eta(x, y) \end{cases}$ виконуються
такі граничні умови:

$$\xi|_{AD} = 0, \quad \xi|_{BC} = a, \quad \eta|_{AB} = 0, \quad \eta|_{CD} = b, \quad (4)$$

$$g_i(x, y) = 0, \quad (x, y) \in \Gamma_i, \quad i = \overline{1, 4}, \quad (5)$$

де $g_i(x, y) = 0, i = \overline{1, 4}$ - рівняння меж області G .

Крім того, в околі межі області виконуються
додаткові умови «зв'язку» межових точок з
внутрішніми у вигляді, наприклад, умов
ортогональності ізоліній відображення до межі
області

$$\text{grad} \xi \cdot \text{grad} \eta = 0. \quad (6)$$

Обернене конформне відображення $G_\xi \rightarrow G_z$
здійснюється такою аналітичною функцією:

$$z(\zeta) = x(\xi, \eta) + i \cdot y(\xi, \eta), \quad (7)$$

дійсна і уявна частина якої задовольняє умовам
Коші-Рімана

$$\frac{\partial x}{\partial \xi} = \frac{\partial y}{\partial \eta}, \quad \frac{\partial x}{\partial \eta} = -\frac{\partial y}{\partial \xi}. \quad (8)$$

На межі параметричного прямокутника
координати межових точок зв'язані рівняннями
межі

$$g_i(x(\xi, \eta), y(\xi, \eta)) = 0, \quad (\xi, \eta) \in \Gamma'_i, \quad i = \overline{1, 4}. \quad (9)$$

Крім того, виконуються умови ортогональності
ізоліній сітки до межі області в такому вигляді:

$$\frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} = 0.$$

Обчислювальний алгоритм побудови
числового конформного відображення
криволінійної області G на параметричний

прямокутник базується на мінімізації
функціоналу Рімана []

$$\hat{O}(\bar{\xi}, \bar{\eta}, M) = \iint_{G_z} \left[\frac{\bar{\xi}^2}{M} + M \cdot \bar{\eta}_x^2 + \frac{\bar{\xi}^2}{M} + M \cdot \bar{\eta}_y^2 \right] dx dy,$$

де $M = \frac{a}{b}$ - модуль параметричного
прямокутника.

З теорії конформних відображень відомо, що
існує невироджене однолисте конформне
відображення $\zeta(z) = \xi(x, y) + i \cdot \eta(x, y)$ області
 G_z на прямокутник G_ξ з деяким певним

відношенням сторін $M = \frac{a}{b}$, яке називається
модулем прямокутника, тобто існують ξ, η, M
такі, що

$$\begin{aligned} \xi_x &= \eta_y, \quad \xi_y = -\eta_x, \\ \xi &\in \left[0, \frac{1}{\sqrt{M}} \right], \quad \eta \in \left[0, \sqrt{M} \right]. \end{aligned} \quad (10)$$

При цьому функції

$$\bar{\xi}(x, y) = \sqrt{M} \cdot \xi(x, y), \quad \bar{\eta}(x, y) = \frac{1}{\sqrt{M}} \cdot \eta(x, y)$$

надають мінімум вищенаведеному функціоналу.

Рівняння Ейлера, які виражають необхідну
умову існування екстремуму функціоналу, в
даному випадку мають вигляд

$$\begin{cases} \bar{\xi}_{xx} + \bar{\eta}_{yy} = 0, \\ \bar{\xi}_{yy} + \bar{\eta}_{xx} = 0, \end{cases} \quad (11)$$

причому

$$M^2 = \frac{\iint_{G_z} (\bar{\xi}_x^2 + \bar{\xi}_y^2) dx dy}{\iint_{G_z} (\bar{\eta}_x^2 + \bar{\eta}_y^2) dx dy}. \quad (12)$$

В силу регулярності відображення $\zeta(z)$ з
якобіаном $J > 0$ існує обернене конформне
відображення

$$z(\bar{\zeta}) = x(\bar{\xi}, \bar{\eta}) + i \cdot y(\bar{\xi}, \bar{\eta}),$$

дійсна і уявна частина яког $x(\bar{\xi}, \bar{\eta}), y(\bar{\xi}, \bar{\eta})$
задовольняють системі нелінійних рівнянь

$$\begin{cases} \alpha \cdot x_{\bar{\xi}\bar{\xi}} - 2\beta \cdot x_{\bar{\xi}\bar{\eta}} + \gamma \cdot x_{\bar{\eta}\bar{\eta}} = 0, \\ \alpha \cdot y_{\bar{\xi}\bar{\xi}} - 2\beta \cdot y_{\bar{\xi}\bar{\eta}} + \gamma \cdot y_{\bar{\eta}\bar{\eta}} = 0, \end{cases} \quad (13)$$

де $\alpha = \frac{x_{\bar{\eta}}^2 + y_{\bar{\eta}}^2}{J_1}$, $\beta = \frac{x_{\bar{\xi}}x_{\bar{\eta}} + y_{\bar{\xi}}y_{\bar{\eta}}}{J_1}$, $\gamma = \frac{x_{\bar{\xi}}^2 + y_{\bar{\xi}}^2}{J_1}$,
 $J_1 = x_{\bar{\xi}}y_{\bar{\eta}} - x_{\bar{\eta}}y_{\bar{\xi}}$ – якобіан оберненого
конформного відображення.

В змінних $\bar{\xi}, \bar{\eta}$ формула (12) для модуля
параметричного прямокутника прийме вигляд

$$M^2 = \frac{\iint_{G_z} \alpha d\bar{\xi} d\bar{\eta}}{\iint_{G_z} \gamma d\bar{\xi} d\bar{\eta}} = \frac{\int_0^1 d\bar{\eta} \int_0^1 \alpha J_1(\bar{\xi}, \bar{\eta}) d\bar{\xi}}{\int_0^1 d\bar{\eta} \int_0^1 \gamma J_1(\bar{\xi}, \bar{\eta}) d\bar{\xi}}.$$

З урахуванням умов ортогональності ліній сітки
 $\beta \equiv 0$ і система рівнянь (13) запишеться так:

$$\begin{cases} \alpha \cdot x_{\bar{\xi}\bar{\xi}} + \gamma \cdot x_{\bar{\eta}\bar{\eta}} = 0, \\ \alpha \cdot y_{\bar{\xi}\bar{\xi}} + \gamma \cdot y_{\bar{\eta}\bar{\eta}} = 0. \end{cases} \quad (14)$$

Система нелінійних рівнянь (14) є
генеруючою і використовується для розрахунку
координат внутрішніх вузлів конформної
різничевої сітки.

Провівши дискретизацію системи (14) та
лінеаризувавши її на деякому кроці, отримуємо

$$\begin{cases} \alpha^{(s-1)} \cdot x_{\bar{\xi}\bar{\xi}}^{(s)} + \gamma^{(s-1)} \cdot x_{\bar{\eta}\bar{\eta}}^{(s)} = 0 \\ \alpha^{(s-1)} \cdot y_{\bar{\xi}\bar{\xi}}^{(s)} + \gamma^{(s-1)} \cdot y_{\bar{\eta}\bar{\eta}}^{(s)} = 0 \end{cases}, \quad (15)$$

де s – номер зовнішньої ітерації.

Розв'язок даної системи для пари координат
відображення x і y отримуємо методом
послідовної верхньої релаксації

$$u_{ij}^{(k+1)} = \omega \cdot \hat{u}_{ij}^{(k)} + (1 - \omega) \cdot u_{ij}^{(k)}, \quad (16)$$

де

$$\hat{u}_{ij}^{(k+1)} = \frac{1}{2(\alpha_1 + \gamma_1)} [\alpha_1(u_{i-1,j}^{k+1} + u_{i+1,j}^k) + \gamma_1(u_{i,j-1}^{k+1} + u_{i,j+1}^k)],$$

$$\omega = \frac{2}{1 + \sqrt{1 - \rho^2}}, \quad \rho = \rho_5 = 1 - 2 \left(\sin^2 \frac{\pi}{2n} + \sin^2 \frac{\pi}{2m} \right),$$

$$\alpha_1 = (x_{i,j+1}^{s-1} - x_{i,j-1}^{s-1})^2 + (y_{i,j+1}^{s-1} - y_{i,j-1}^{s-1})^2, \quad (17)$$

$$\gamma_1 = (x_{i+1,j}^{s-1} - x_{i-1,j}^{s-1})^2 + (y_{i+1,j}^{s-1} - y_{i-1,j}^{s-1})^2. \quad (18)$$

Тут під u розуміються координати відображення
 x та y .

Одним з обов'язкових етапів побудови
конформної ортогональної різничевої сітки є
уточнення вузлів, які знаходяться на межі
області і в ході роботи алгоритму «плавають» по
ній. Перерахунок координат здійснюється

наступним чином: одна координата
обчислюється з дискретизації умов Коші-Рімана
або умов ортогональності, з рівняння межі за
однією відомою координатою.

Запишемо в загальному вигляді різницеві
співвідношення для визначення координат
"плаваючих" вузлів, які потрібно доповнити
умовами (7), (8) для відповідних меж. Зокрема,
матимемо:

для межі Γ_1

$$(x_{i+1,0} - x_{i-1,0})(-x_{i2} + 4x_{i1} - 3x_{i0}) + (y_{i+1,0} - y_{i-1,0})(-y_{i2} + 4y_{i1} - 3y_{i0}) = 0, \quad i = \overline{1, n-1};$$

для межі Γ_2

$$(x_{i+1,m} - x_{i-1,m})(x_{i,m} - 4x_{i,m-1} + 3x_{i,m-2}) + (y_{i+1,m} - y_{i-1,m})(y_{i,m} - 4y_{i,m-1} + 3y_{i,m-2}) = 0, \quad i = \overline{1, n-1};$$

для межі Γ_3

$$(-x_{2j} + 4x_{1j} - 3x_{0j})(x_{0,j+1} - x_{0,j-1}) + (-y_{2j} + 4y_{1j} - 3y_{0j})(y_{0,j+1} - y_{0,j-1}) = 0, \quad j = \overline{1, m-1};$$

для межі Γ_4

$$(x_{n-2j} - 4x_{n-1j} + 3x_{nj})(x_{n,j+1} - x_{n,j-1}) + (y_{n-2j} - 4y_{n-1j} + 3y_{nj})(y_{n,j+1} - y_{n,j-1}) = 0, \quad j = \overline{1, m-1}.$$

Для перевірки отриманої різничевої сітки на
«конформність», використовуємо критерій
конформності різничевої сітки за розрахованими
вузлами сітки

$$k = \frac{1}{n \cdot (m+1)} \sum_{i=0}^{n-1} \left[\sum_{j=0, j \neq M0}^m \left| \frac{d_{ij}^1}{d_{ij}^2} - 1 \right| \right], \quad (19)$$

$$\text{де} \quad d_{ij}^1 = \sqrt{(x_{ij} - x_{i+1,j+1})^2 + (y_{ij} - y_{i+1,j+1})^2},$$

$$d_{ij}^2 = \sqrt{(x_{i,j+1} - x_{i+1,j})^2 + (y_{i,j+1} - y_{i+1,j})^2}.$$

Використовуючи метод послідовної верхньої
релаксації для уточнення координат внутрішніх
вузлів різничевої сітки та уточнення координат
«плаваючих вузлів» на межі області, а також
уточнення модуля області (зміни числа розбиттів
сторін параметричного прямокутника) в
результаті чергування ітераційних процесів
уточнення координат внутрішніх вузлів і
координат «плаваючих» вузлів, розраховуємо
координати внутрішніх і зовнішніх вузлів
конформної різничевої сітки у вигляді двох
масивів (x_{ij}, y_{ij}) , $i = \overline{0, n}$, $j = \overline{0, m}$, які є
прообразами відповідних їм вузлів (ξ_i, η_j) ,
 $i = \overline{0, n}$, $j = \overline{0, m}$ параметричного прямокутника.

4.2. Математична модель задачі (1),(2) в змінних ξ, η області параметричного прямокутника

З умов Коші-Рімана для оберненого
конформного відображення

$$x_\xi = y_\eta, \quad x_\eta = -y_\xi$$

маємо наступний зв'язок між першими
похідними

$$\xi_x = \frac{y_\eta}{J_1}, \quad \eta_x = \frac{-y_\xi}{J_1}, \quad \xi_y = \frac{-x_\eta}{J_1}, \quad \eta_y = \frac{x_\xi}{J_1}.$$

На основі цього, аналогічно знаходимо формули,
які виражають зв'язок між другими похідними
прямого і оберненого конформного
відображення.

Після переходу в основній математичній
моделі задачі (1),(2) до змінних ξ, η з
використанням вищезгаданих формул,
отримаємо її запис в змінних ξ, η області
параметричного прямокутника.

$$D\left(\frac{\partial^2 c}{\partial \xi^2} + \frac{\partial^2 c}{\partial \eta^2}\right) = -Q \cdot J(\xi, \eta) \delta(\xi - \xi_0) \delta(\eta - \eta_0), \quad (19)$$

$$\tilde{n}(\xi, 0) = c(\xi, b) = c(0, \eta) = c(a, \eta) = 0, \quad (20)$$

де $J(\xi, \eta) = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi}$ - якобіан оберненого

конформного відображення; ξ_0, η_0 - координати
образа джерела забруднення в параметричному
прямокутнику G_ξ .

4.3. Знаходження координат місцеположення джерела методом скінченних синус-перетворень Фур'є на основі розв'язку системи нелінійних рівнянь

Застосувавши до рівняння (19) скінченне
синус - перетворення Фур'є по змінних x і y
при відомих координатах точки розміщення
джерела (x_0, y_0) знаходимо функцію Гріна

$$\tilde{n}(\xi, \eta) = \frac{Q \cdot J(\xi_0, \eta_0)}{D\pi^2 ab} \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} \frac{\sin \frac{n\pi\xi}{a} \sin \frac{m\pi\eta}{b}}{\left(\frac{n}{a}\right)^2 + \left(\frac{m}{b}\right)^2} \cdot \sin \frac{n\pi\xi_0}{a} \sin \frac{m\pi\eta_0}{b}.$$

Нехай відомі значення концентрації
забруднень відповідно в точках $(\xi_1, \eta_1), (\xi_2, \eta_2)$

$$(\xi_1, \eta_1) : c(\xi_1, \eta_1, x_0, y_0) = \bar{C}_1,$$

$$(\xi_2, \eta_2) : c(\xi_2, \eta_2, x_0, y_0) = \bar{C}_2.$$

Тоді для знаходження координати точки
 $M(\xi_0, \eta_0)$, отримаємо систему двох нелінійних
рівнянь з двома невідомими

$$f_1(x_0, y_0) \equiv J(\xi_0, \eta_0) \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} \frac{\sin \frac{n\pi\xi_1}{a} \sin \frac{m\pi\eta_1}{b}}{\left(\frac{n}{a}\right)^2 + \left(\frac{m}{b}\right)^2} \cdot$$

$$\cdot \sin \frac{n\pi\xi_0}{a} \sin \frac{m\pi\eta_0}{b} - \frac{\bar{C}_1 \pi^2 ab}{Q} = 0,$$

$$f_2(x_0, y_0) \equiv J(\xi_0, \eta_0) \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} \frac{\sin \frac{n\pi\xi_2}{a} \sin \frac{m\pi\eta_2}{b}}{\left(\frac{n}{a}\right)^2 + \left(\frac{m}{b}\right)^2} \cdot$$

$$\cdot \sin \frac{n\pi\xi_0}{a} \sin \frac{m\pi\eta_0}{b} - \frac{\bar{C}_2 \pi^2 ab}{Q} = 0.$$

Розв'язавши дану систему одним із числових
методів [13], отримаємо наближені значення
координат (ξ_0, η_0) точки розміщення джерела
забруднення в області параметричного
прямокутника. Тоді з оберненого конформного
відображення знайдемо координати (x_0, y_0)
джерела забруднень в фізичній області.

4.4 Розв'язок задачі з використанням інтегральних рівнянь

Замість (19), (20), згідно [7] будемо
розв'язувати наступну задачу:

$$Lc(\xi, \eta) = f(\xi, \eta), \quad (21)$$

$$c(\xi, \eta)|_{\bar{A}} = 0, \quad (22)$$

де $f(\xi, \eta)$ - поки-що невідома функція. Тоді
розв'язок задачі (21), (22) представляється у
такому вигляді:

$$\tilde{n}(\xi, \eta) = \int_0^a d\psi \int_0^b G(\xi, \eta, \psi, \zeta) f(\psi, \zeta) d\zeta, \quad (23)$$

де функція Гріна $G(\xi, \eta, \psi, \zeta)$ в даному випадку
має представлення

$$G(\xi, \eta, \psi, \zeta) = \frac{Q \cdot J(\psi, \zeta)}{D\pi^2 ab} \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} \frac{\sin \frac{n\pi\xi}{a} \sin \frac{m\pi\eta}{b}}{\left(\frac{n}{a}\right)^2 + \left(\frac{m}{b}\right)^2} \cdot \sin \frac{n\pi\psi}{a} \sin \frac{m\pi\zeta}{b}.$$

Нехай в точках $M_i(x_i, y_i)$, $i = \overline{1, N}$ відомо
 N замірив концентрації $\bar{C}_i$. Тоді для

знаходження невідомої функції $f(\xi, \eta)$ маємо систему N інтегральних рівнянь

$$\bar{C}_i = c(\xi_i, \eta_i) = \int_0^a d\psi \int_0^b G(\xi_i, \eta_i, \psi, \zeta) f(\psi, \zeta) d\zeta, i = \overline{1, N}.$$

Розв'язок даної системи шукаємо у вигляді

$$\hat{f}(\psi, \zeta) = \sum_{j=1}^N G(\psi, \zeta, \xi_j, \eta_j) b_j,$$

де $b_j, j = \overline{1, N}$ - невідомі коефіцієнти, які знаходимо із розв'язку наступної системи лінійних алгебричних рівнянь:

$$\sum_{i=1}^N a_{ij} \cdot b_j = \bar{C}_i, i, j = \overline{1, N}, \quad (24)$$

коефіцієнти якої $a_{ij}, i, j = \overline{1, N}$ є відомі і обчислюються за формулами

$$a_{ij} = \int_0^a d\psi \int_0^b G(\xi_i, \eta_i, \psi, \zeta) G(\psi, \zeta, \xi_j, \eta_j) d\zeta, i, j = \overline{1, N} J_i = \tilde{N}_i = QC_i^*(x_0, y_0) = \frac{J(\psi, \zeta)}{D\pi^2 ab} \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} \frac{\sin \frac{n\pi \xi_i}{a} \sin \frac{m\pi \eta_i}{b}}{\left(\frac{n}{a}\right)^2 + \left(\frac{m}{b}\right)^2} \cdot \sin \frac{n\pi \xi_0}{a} \sin \frac{m\pi \eta_0}{b},$$

Система лінійних алгебричних рівнянь (24) має єдиний розв'язок b_j , оскільки матриця $A = \{a_{ij}\}, i, j = \overline{1, N}$ є додатньовизначена. Тоді координати місцеположення точки $M_0(\xi_0, \eta_0)$ розміщення джерела забруднення визначаються як:

$$(\xi_0, \eta_0) = (\hat{\psi}, \hat{\zeta}) = \text{Arg max } \hat{f}(\hat{\psi}, \hat{\zeta}).$$

На основі побудованого оберненого конформного відображення знаходимо координати (x_0, y_0) .

4.5 Розв'язок задачі з використанням спряжених рівнянь

Спряжена задача до (21), (22) згідно [6] отримується наступним чином. Рівняння (19) домножаємо на деяку функцію $C^* = C^*(\xi, \eta)$ і інтегруємо по всій області параметричного прямокутника

$$\int_0^a \int_0^b C^* D \left(\frac{\partial^2 c}{\partial \xi^2} + \frac{\partial^2 c}{\partial \eta^2} \right) d\xi d\eta = -Q \int_0^a \int_0^b C^* J(\xi, \eta) \delta(\xi - \xi_0) \delta(\eta - \eta_0) d\xi d\eta.$$

Перетворимо ліву частину так, щоб функція $\tilde{N}^*$ стояла під знаком диференціала

$$\int_0^a \int_0^b D c \left(\frac{\partial^2 C^*}{\partial \xi^2} + \frac{\partial^2 C^*}{\partial \eta^2} \right) d\xi d\eta + \int_0^a \left(C^* \frac{\partial c}{\partial \xi} - \frac{\partial^2 C^*}{\partial \xi^2} \right) d\eta \Big|_0^b + \int_0^b \left(C^* \frac{\partial c}{\partial \eta} - \frac{\partial^2 C^*}{\partial \eta^2} \right) d\xi \Big|_0^a = -Q \int_0^a \int_0^b C^* J(\xi, \eta) \delta(\xi - \xi_0) \delta(\eta - \eta_0) d\xi d\eta.$$

Припустимо, що функція $\tilde{N}^*$ задовольняє таким спряженим рівнянням:

$$D \left(\frac{\partial^2 \tilde{N}^*}{\partial \xi^2} + \frac{\partial^2 \tilde{N}^*}{\partial \eta^2} \right) = P_i, i = \overline{1, N}.$$

Лінійний функціонал J_i від концентрації C^* в даному випадку має вигляд

$$\tilde{N}_i^*(\xi, \eta) - \text{розв'язки спряжених задач з } P_i, i = \overline{1, N} \text{ правою частиною; } \tilde{N}_i = \bar{C}(\xi_i, \eta_i), i = \overline{1, N} - \text{виміряні значення концентрації в точках } (x(\xi_i, \eta_i), y(\xi_i, \eta_i)).$$

Знайшовши координати (ξ_0, η_0) , місцеположення джерела в параметричному прямокутнику застосувавши обернене конформне відображення знаходимо координати (x_0, y_0) місцеположення джерела забруднень в фізичній області.

5. Результати числових експериментів. Виконана програмна реалізація розв'язку даної задачі в середовищі Delphi 7.0, на основі чого проведені числові експерименти та зроблено їх аналіз. В якості тестового прикладу розглянуто джерело забруднення, розміщене в центрі круга $x_0 = 0, y_0 = 0$ радіуса $r = 10$.

Здійснимо числове конформне відображення даного круга на квадрат згідно алгоритму [1] (рис.3).

При $D = 1, Q = 1$ отримаємо наступний розподіл концентрації забруднення від даного джерела (рис.4), розміщеного в фізичній області.

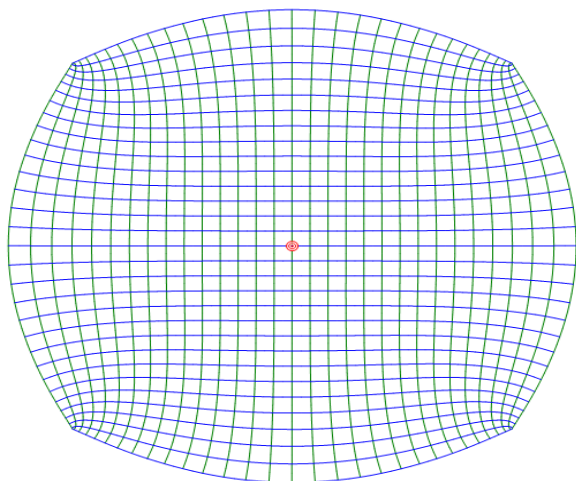


Рис.3. Джерело забруднення в крузі

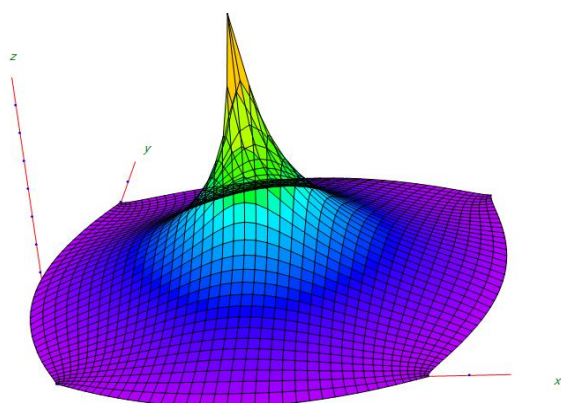


Рис.4. Розподіл концентрації забруднень від джерела, розміщеного в крузі в точці в $M(x_0, y_0)$

З числових розрахунків прямої задачі взято значення концентрації $\tilde{N}_1 = 0,05$ в точці $x_1 = 4$, $y_1 = 4$ і $\tilde{N}_2 = 0,03$ в точці $x_2 = 7$, $y_2 = 7$. При тих же вхідних даних знайдено наближене значення координат розміщення джерела забруднення трьома методами. Результати числових розрахунків представлені в таб.1.

Таблиця 1

Точний розв'язок		Алгоритм №1		Алгоритм №2		Алгоритм №3	
x_0	y_0	x_0	y_0	x_0	y_0	x_0	y_0
5	5	4.84	4.88	4.79	4.81	4.93	4.98
Похибки		0.16	0.12	0.21	0.19	0.07	0.02

В результаті проведених числових експериментів показано достатньо добре співпадіння результатів розрахунків різними методами.

7. Висновки. В роботі знайдено невідомі параметри точкового джерела забруднення, розміщеного в однозв'язній області з криволінійною межею для стаціонарного випадку. Отримано числові розв'язки даної задачі трьома алгоритмами (методом функції Гріна, інтегральних рівнянь, які зводяться до системи лінійних алгебраїчних рівнянь та методом спряжених рівнянь). Розроблені алгоритми отримання числових розрахунків програмно реалізовані з використанням сучасних технологій програмування, на основі чого проведені числові експерименти та здійснено їх аналіз. Показано достатньо добре співпадіння результатів розрахунків, обчислених різними методами.

Список використаних джерел

1. Власюк А.П. Локалізація місцеположення джерела забруднення в двовимірних нестационарних задачах масо переносу / А.П. Власюк, О.М. Багнюк // Тез. доп. XIX міжнар. конф. «Проблеми прийняття рішень в умовах невизначеності», – Київ. – 2012. – С.67-68.
2. Власюк А.П. Ідентифікація місцеположення джерела забруднення в стаціонарній одновимірній задачі масоперенесення / А.П. Власюк, О.М. Багнюк // Матем. та комп. моделювання. Серія «Технічні науки». – 2012. – Вип. 6. – С. 40–48.
3. Власюк А.П. Знаходження місцеположення джерела забруднення в одній двовимірній стаціонарній задачі масопереносу / А.П. Власюк, О.М. Багнюк // XIV міжнар. конф. ім. акад. М.Кравчука. –К. – 2012. – С. 107–108.
4. Власюк А.П. Автоматическое построение конформных и квазиконформных отображений двух- и трехсвязных областей / А.П. Власюк, В.Г. Михальчук // (Препр. №2), - Киев, 1991. – 56 с
5. Власюк А.П. Автоматическое построение конформных и квазиконформных отображений четырехугольных областей с помощью разностных сеток с “плавающими узлами” / А.П. Власюк, В.Г. Михальчук // препр. АНУССР. Ин-т математики; 89.79. – К.:1989 г. – 55с.

6. Марчук Г.И. Математическое моделирование в проблеме окружающей среды. – М.: Наука, – 1982. – 320 с.
7. Наконечний О.Г., Оптимальне керування та оцінювання в рівняннях із частинними похідними: К.: «Київський університет». – 2004. – 103 с.
8. Снеддон И. Преобразования Фурье М.: ИЛ. – 1955. – 668 с.
9. Борухов В.Т. Численное решение обратной задачи восстановления источника в параболическом уравнении / В.Т. Борухов, П.Н. Вабищевич // Математическое моделирование. – 1998. – Т. 10, №11, - С. 93-100.
10. Стеля О.Б. Чисельна процедура відтворення невідомих параметрів джерел для задач масопереносу / О.Б. Стеля, О.С. Тригуб // Національний університет “Києво-Могилянська академія”, Наукові записки, – 2000. – Т. 18, Комп’ютерні науки. – С. 40–46.
11. Криксин Ю.А. Обратная задача восстановления источника для уравнения конвективной диффузии / Ю.А. Криксин, С.Н. Плющев, Е.А. Самарская, В.Ф. Тишкин // Математическое моделирование №11. – С.93 – 100 .
12. Ортега Дж. Итерационные методы решения нелинейных систем уравнений со многими неизвестными / Дж. Ортега, В. Рейнболдт// – М.: Мир, 1975. – 558 с.
13. Алоян А.Е. Математическое моделирование в проблеме окружающей среды / А.Е. Алоян, В.В. Пененко, В.В. Козодеров// Современные проблемы вычислительной математики и математического моделирования. – М.: Наука, 2005. – Т. 2. – С. 277 –349.
14. Панасенко А.Е. Численное решение некоторых обратных задач переноса примеси на многопроцессорных вычислительных системах / А.Е. Панасенко, А.В. Старченко// Четвертая Сибирская школа-семинар по параллельным высокопроизводительным вычислениям. – Томск.: Дельтаплан, 2008. – С. 139 –148.
15. Пененко В.В. Модели и методы для охраны окружающей среды / В.В. Пененко, Е.А. Цветова// Оптика атмосферы и океана. 2002. – Т. 15. – № 5/6. – С. 412 – 418.

Надійшла до редколегії 05.09.2013

УДК 004.655

Глушко* І.М.

Розширена мультимножинна таблична алгебра: операції з'єднання

Мультимножинна таблична алгебра поповнена новими операціями: операціями внутрішніх і зовнішніх з'єднань, операцією напівз'єднання. Задано формальну математичну семантику вказаних операцій. Для задання зовнішніх з'єднань введено особливий елемент універсального домену *NULL*.

Ключові слова: реляційні бази даних, мультимножинна таблична алгебра, розширена мультимножинна таблична алгебра.

Ніжинський державний університет імені
Миколи Гоголя, 16600, м. Ніжин, вул.
Кропив'янського 2,
e-mail: glushkoim@gmail.com

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

Вступ

Реляційна модель даних в теперішній час широко використовується як у наукових дослідженнях в базах даних, так на практиці. Дана модель базується на множинах кортежів, тобто не дозволяє дублювати кортежів у відношенні [1]. Проте багато мов, орієнтованих на роботу з базами даних, вимагають реляційну модель даних з мультимножинною семантикою, що передбачає розуміння таблиць як мультимножин, тобто сукупностей з дублікатами. Питанню використання мультимножин в базах даних приділяли увагу Paul W.P.J. Grefen та Rolf A. de By [2], G. Lamperti, M. Melchiori, M. Zanella [3], A. Silbeschatz, H. Korth, S. Sudarshan [4], Д.Б. Буй, С.А. Поляков, Ю.Й. Брона, В.Н. Редько [5]. У [2-4] вводиться до розгляду мультимножинна таблична алгебра та задаються її операції. Разом з тим, в жодній із зазначених робіт не приділяється увага операціям внутрішніх та зовнішніх з'єднань над таблицями мультимножинної табличної алгебри.

I.M. Glushko*

Extending multiset table algebra: join operations

Multiset table algebra is filled up with new operations such as inner and outer joins, semijoin. A formal mathematical semantics of these operations is defined. The special element *NULL* is inserted in the universal domain for to define of outer join.

Key words: relation databases, multiset table algebra, extending multiset table algebra.

Nizhyn Gogol State University, 16600, Nizhyn,
Kropyiv'yanskoho str, 2,
e-mail: glushkoim@gmail.com

Мультимножинна таблична алгебра

Всі невизначені тут поняття розуміємо в сенсі робіт [5, 6]. Розглянемо дві множини: A – множину атрибутів і D – універсальний домен.

Довільну скінченну множину атрибутів $R \subseteq A$ назовемо схемою.

Рядком схеми R називається іменна множина на парі R, D , проекція якої за першою компонентою рівна R .

Під таблицею розуміємо пару $\langle \psi, R \rangle$, де перша компонента ψ – це довільна мультимножина, зокрема, нескінченна, а друга компонента R – схема таблиці.

Під мультимножинною табличною алгеброю розуміємо алгебру $\langle \Psi, \Omega_{P,\Xi} \rangle$, де $\Psi = \bigcup_{R \subseteq A} \Psi(R)$ –

множина усіх таблиць, $\Psi(R)$ – множина усіх таблиць схеми R ,

$\Omega_{P,\Xi} = \{ \bigcup_{All}^R, \bigcap_{All}^R, \setminus_{All}^R, \sigma_{p,R}, \pi_{X,R}, \otimes_{R_1,R_2}, Rt_{\xi,R}, \sim_R \}$ –

сигнатура, $p \in P, \xi \in \Xi, X, R, R_1, R_2 \subseteq A, P, \Xi$ – множини параметрів. Операції сигнатури мультимножинної табличної алгебри задано в [6].

Позначимо через $Occ(s, \psi)$ – кількість дублікатів (екземплярів) рядка s у

мультимножині ψ . Доведимо мультимножину ψ записувати як $\{s_1^{n_1}, \dots, s_k^{n_k}, \dots\}$, де $n_i = \text{Occ}(s_i, \psi)$, $i = 1, 2, \dots$, а $\Theta(\psi) = \{s_1, \dots, s_k, \dots\}$ – основа мультимножин ψ .

Поповнимо сигнатуру мультимножинної табличної алгебри операціями внутрішніх і зовнішніх з'єднань та операцією напівз'єднання.

Операції внутрішнього з'єднання

Під декартовим з'єднанням таблиць схем R_1 та R_2 , причому $R_1 \cap R_2 = \emptyset$, розуміється бінарна параметрична операція вигляду $Cj_{R_1, R_2} : \Psi(R_1) \times \Psi(R_2) \rightarrow \Psi(R_1 \cup R_2)$, причому

$$\langle \psi_1, R_1 \rangle Cj_{R_1, R_2} \langle \psi_2, R_2 \rangle = \langle \psi', R_1 \cup R_2 \rangle, \quad \text{де}$$

$$\langle \psi_1, R_1 \rangle \in \Psi(R_1), \quad \langle \psi_2, R_2 \rangle \in \Psi(R_2).$$

Основою мультимножини ψ' є множина рядків $\Theta(\psi') = \{s \mid \exists s_1 \exists s_2 (s_1 \in \Theta(\psi_1) \wedge s_2 \in \Theta(\psi_2) \wedge s = s_1 \cup s_2)\}$.

Кількість дублікатів знаходиться так: $\text{Occ}(s, \psi') = \text{Occ}(s_1, \psi_1) \cdot \text{Occ}(s_2, \psi_2)$, де $s \in \Theta(\psi')$ і $s_1 = s \mid R_1$, $s_2 = s \mid R_2$ – обмеження рядка s на вказані схеми (див., наприклад, [5]).

Під внутрішнім природним з'єднанням таблиць схем R_1 та R_2 розуміється бінарна параметрична операція $\otimes_{R_1, R_2}$, значеннями якої є таблиці схеми $R_1 \cup R_2$, що складаються з усяких об'єднань сумісних рядків вихідних таблиць.

Отже, $\otimes_{R_1, R_2} : \Psi(R_1) \times \Psi(R_2) \rightarrow \Psi(R_1 \cup R_2)$, де $\langle \psi_1, R_1 \rangle \otimes_{R_1, R_2} \langle \psi_2, R_2 \rangle = \langle \psi', R_1 \cup R_2 \rangle$, в припущенні $\langle \psi_1, R_1 \rangle \in \Psi(R_1)$, $\langle \psi_2, R_2 \rangle \in \Psi(R_2)$.

Основою мультимножини ψ' є множина рядків $\Theta(\psi') = \{s \mid \exists s_1 \exists s_2 (s_1 \in \Theta(\psi_1) \wedge s_2 \in \Theta(\psi_2) \wedge s_1 \approx s_2 \wedge s = s_1 \cup s_2)\}$, де $\approx$ – відношення сумісності рядків (див., наприклад, [5]).

Кількість дублікатів знаходиться так: $\text{Occ}(s, \psi') = \text{Occ}(s_1, \psi_1) \cdot \text{Occ}(s_2, \psi_2)$, де, як і раніше, $s \in \Theta(\psi')$ і $s_1 = s \mid R_1$, $s_2 = s \mid R_2$.

Під внутрішнім з'єднанням за атрибутами $A_1, \dots, A_n$, причому всі $A_1, \dots, A_n$ попарно різні, $n \geq 1$, таблиць схем R_1 та R_2 , де $R_1 \cap R_2 = \{A_1, \dots, A_n\}$ розуміється бінарна параметрична операція вигляду

$$\otimes_{A_1, \dots, A_n, R_1, R_2} : \Psi(R_1) \times \Psi(R_2) \rightarrow \Psi(R_1 \cup R_2), \quad \text{де}$$

$$\langle \psi_1, R_1 \rangle \otimes_{A_1, \dots, A_n, R_1, R_2} \langle \psi_2, R_2 \rangle = \langle \psi', R_1 \cup R_2 \rangle, \quad \text{в}$$

припущенні $\langle \psi_1, R_1 \rangle \in \Psi(R_1)$, $\langle \psi_2, R_2 \rangle \in \Psi(R_2)$.

Основою мультимножини ψ' є множина рядків $\Theta(\psi') = \{s \mid \exists s_1 \exists s_2 (s_1 \in \Theta(\psi_1) \wedge s_2 \in \Theta(\psi_2) \wedge \bigwedge_{i=1}^n s_1(A_i) = s_2(A_i) \wedge s = s_1 \cup s_2)\}$.

Кількість дублікатів знаходиться так: $\text{Occ}(s, \psi') = \text{Occ}(s_1, \psi_1) \cdot \text{Occ}(s_2, \psi_2)$, де $s \in \Theta(\psi')$ і $s_1 = s \mid R_1$, $s_2 = s \mid R_2$.

Зауважимо, що при внутрішньому з'єднанні за атрибутами $A_1, \dots, A_n$ перетин схем таблиць-аргументів, взагалі кажучи, включає перелічені атрибути. Якщо таблиці-аргументи мають ще й інші спільні атрибути, які відрізняються від атрибутів $A_1, \dots, A_n$, то перед з'єднанням їх потрібно перейменувати.

Нехай $p : S \times S \rightarrow \{true, false\}$ – частковий бінарний предикат на множині всіх рядків S , такий, що $\forall s_1 \forall s_2 (\langle s_1, s_2 \rangle \in \text{dom } p \wedge p(s_1, s_2) = true \Rightarrow s_1 \approx s_2)$.

Під внутрішнім з'єднанням за предикатом p таблиць схем R_1 та R_2 розуміється часткова бінарна параметрична операція вигляду $\otimes_{p, R_1, R_2} : \Psi(R_1) \times \Psi(R_2) \rightarrow \Psi(R_1 \cup R_2)$, де

$$\text{dom } \otimes_{p, R_1, R_2} = \{\langle \langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle \rangle \mid \Theta(\psi_1) \times \Theta(\psi_2) \subseteq \text{dom } p\} \quad \text{та}$$

$$\langle \psi_1, R_1 \rangle \otimes_{p, R_1, R_2} \langle \psi_2, R_2 \rangle = \langle \psi', R_1 \cup R_2 \rangle.$$

Основою мультимножини ψ' є множина рядків $\Theta(\psi') = \{s \mid \exists s_1 \exists s_2 (s_1 \in \Theta(\psi_1) \wedge s_2 \in \Theta(\psi_2) \wedge p(s_1, s_2) = true \wedge s = s_1 \cup s_2)\}$.

Кількість дублікатів знаходиться так: $\text{Occ}(s, \psi') = \text{Occ}(s_1, \psi_1) \cdot \text{Occ}(s_2, \psi_2)$, де $s \in \Theta(\psi')$ і $s_1 = s \mid R_1$, $s_2 = s \mid R_2$.

Зрозуміло, що пара таблиць $\langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle$ належить вказаній області означеності.

Операція природного з'єднання $\otimes_{R_1, R_2}$ є розширенням довільної іншої операції з'єднання в наступному розумінні:

$$\langle \psi_1, R_1 \rangle Cj_{R_1, R_2} \langle \psi_2, R_2 \rangle = \langle \psi_1, R_1 \rangle \otimes_{R_1, R_2} \langle \psi_2, R_2 \rangle,$$

$$\langle \psi_1, R_1 \rangle \otimes_{A_1, \dots, A_n, R_1, R_2} \langle \psi_2, R_2 \rangle = \langle \psi_1, R_1 \rangle \otimes_{R_1, R_2} \langle \psi_2, R_2 \rangle,$$

$$\left(\langle \psi_1, R_1 \rangle \otimes_{p, R_1, R_2} \langle \psi_2, R_2 \rangle \right)_1 \preceq \left(\langle \psi_1, R_1 \rangle \otimes_{R_1, R_2} \langle \psi_2, R_2 \rangle \right)_1,$$

за умови визначення значень операцій у лівих частинах цих двох рівностей та включення ($\preceq$ – відношення включення мультимножин [5]).

Запис $(\varphi(\langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle))_1$, позначає першу компоненту таблиці $\varphi(\langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle)$.

Під операцією напівз'єднання двох таблиць схем R_1 та R_2 розуміється бінарна параметрична операція $\bowtie_{R_1, R_2}$, значенням якої є таблиця схеми R_1 , що містить ті рядки першої таблиці, які входять у (природне) з'єднання таблиць-аргументів.

Таким чином, де $\bowtie_{R_1, R_2} : \Psi(R_1) \times \Psi(R_2) \rightarrow \Psi(R_1)$,
 $\langle \psi_1, R_1 \rangle \bowtie_{R_1, R_2} \langle \psi_2, R_2 \rangle = \langle \psi', R_1 \cup R_2 \rangle$, в
припущенні $\langle \psi_1, R_1 \rangle \in \Psi(R_1)$, $\langle \psi_2, R_2 \rangle \in \Psi(R_2)$.

Основою мультимножини ψ' є множина рядків

$$\Theta(\psi') = \{s_1 \mid \exists s_2 (s_1 \in \Theta(\psi_1) \wedge s_2 \in \Theta(\psi_2) \wedge s_1 \approx s_2)\}.$$

Кількість дублікатів знаходиться так:
 $Occ(s, \psi') = Occ(s, \psi_1)$, де $s \in \Theta(\psi')$.

Операції зовнішнього з'єднання

Для позначення відсутніх значень у результуючій таблиці, використовуємо особливий елемент універсального домену $NULL$. Позначимо через s_R^{NULL} константний рядок схеми R , тобто $s_R^{NULL} : R \rightarrow \{NULL\}$.

Використаємо логічну схему задання операцій зовнішнього з'єднання [5]. Нехай $\varphi : \Psi(R_1) \times \Psi(R_2) \rightarrow \Psi(R_1 \cup R_2)$ – деяка часткова бінарна операція на множині таблиць, причому виконується включення

$$(\varphi(\langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle))_1 \preceq \left(\langle \psi_1, R_1 \rangle \otimes_{R_1, R_2} \langle \psi_2, R_2 \rangle \right)_1$$

для всіх $\langle \langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle \rangle \in \text{dom } \varphi$. Зауважимо, що операції C_j , $\otimes_{R_1, R_2}$, $\otimes_{A_1, \dots, A_n, R_1, R_2}$, $\otimes_{p, R_1, R_2}$ саме такі.

Зафіксуємо таблиці $\langle \psi_1, R_1 \rangle$, $\langle \psi_2, R_2 \rangle$ з області означеності операції φ . Тоді таблиця $\langle \psi_1, R_1 \rangle$ припускає наступне представлення:

$$\langle \psi_1, R_1 \rangle = \left\langle \psi_1 \cap_{\varphi} \psi_2, R_1 \right\rangle \cup_{All}^{R_1} \left\langle \psi_1 - \psi_2, R_1 \right\rangle, \quad \text{де}$$

$$\left\langle \psi_1 \cap_{\varphi} \psi_2, R_1 \right\rangle = \langle \psi', R_1 \rangle, \quad \text{основою мультимножини}$$

ψ' є множина рядків

$$\Theta(\psi') = \{s_1 \mid s_1 \in \Theta(\psi_1) \wedge \exists s_2 (s_2 \in \Theta(\psi_2) \wedge s_1 \cup s_2 \in \Theta((\varphi(\langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle))_1))\}, \quad \text{а кількість}$$

$$\text{дублікатів } Occ(s_1, \psi') = Occ(s_1, \psi_1), \quad \text{де } s_1 \in \Theta(\psi'),$$

та $\left\langle \psi_1 - \psi_2, R_1 \right\rangle = \langle \psi'', R_1 \rangle$, основою

мультимножини ψ'' є множина рядків

$$\Theta(\psi'') = \{s_1 \mid s_1 \in \Theta(\psi_1) \wedge \forall s_2 (s_2 \in \Theta(\psi_2) \Rightarrow s_1 \cup s_2 \notin \Theta((\varphi(\langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle))_1))\}, \quad \text{а кількість}$$

дублікатів знаходиться так:

$$Occ(s_1, \psi'') = Occ(s_1, \psi_1), \quad \text{де } s_1 \in \Theta(\psi'').$$

Іншими словами, рядки з таблиці

$$\left\langle \psi_1 \cap_{\varphi} \psi_2, R_1 \right\rangle \text{ використовуються в формуванні}$$

результату з'єднання, а рядки з таблиці

$$\left\langle \psi_1 - \psi_2, R_1 \right\rangle \text{ не використовуються.}$$

Представлення таблиці $\langle \psi_2, R_2 \rangle$ отримаємо,

замінивши ролі таблиць $\langle \psi_1, R_1 \rangle$ і $\langle \psi_2, R_2 \rangle$ у представленні таблиці $\langle \psi_1, R_1 \rangle$.

Задамо чотири операції зовнішнього з'єднання, індуковані операцією внутрішнього з'єднання φ . Для цього розглянемо наступні природні з'єднання:

$$\left\langle \psi_1 - \psi_2, R_1 \right\rangle \otimes_{R_1, R_2 \setminus R_1} \left\langle \{s_{R_2 \setminus R_1}^{NULL}\}, R_2 \setminus R_1 \right\rangle = \langle \psi', R_1 \cup R_2 \rangle,$$

де основою мультимножини ψ' є множина рядків

$$\Theta(\psi') = \{s_1 \cup s_{R_2 \setminus R_1}^{NULL} \mid s_1 \in \Theta(\psi_1 - \psi_2)\}, \quad \text{а кількість}$$

$$\text{дублікатів } Occ(s', \psi') = Occ(s' \mid R_1, \psi_1 - \psi_2),$$

$s' \in \Theta(\psi')$ і

$$\left\langle \psi_2 - \psi_1, R_2 \right\rangle \otimes_{R_2, R_1 \setminus R_2} \left\langle \{s_{R_1 \setminus R_2}^{NULL}\}, R_1 \setminus R_2 \right\rangle = \langle \psi'', R_1 \cup R_2 \rangle,$$

де основою мультимножини ψ'' є множина рядків

$$\Theta(\psi'') = \{s_{R_1 \setminus R_2}^{NULL} \cup s_2 \mid s_2 \in \Theta(\psi_2 - \psi_1)\}, \quad \text{а}$$

кількість дублікатів

$$Occ(s'', \psi'') = Occ(s'' \mid R_2, \psi_2 - \psi_1), \quad s'' \in \Theta(\psi'').$$

Під зовнішнім лівим з'єднанням, індукованим операцією φ , розуміється часткова бінарна операція вигляду

$\varphi_l : \Psi(R_1) \times \Psi(R_2) \xrightarrow{\sim} \Psi(R_1 \cup R_2)$, де
 $\text{dom } \varphi_l = \text{dom } \varphi$, причому

$$\varphi_l(\langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle) = \varphi(\langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle) \cup_{All}^{R_1 \cup R_2} \langle \psi_1 - \psi_2, R_1 \rangle \otimes_{R_1, R_2 \setminus R_1} \langle \{s_{R_2 \setminus R_1}^{NULL}\}, R_2 \setminus R_1 \rangle.$$

Під зовнішнім правим з'єднанням, індукованим операцією φ , розуміється часткова бінарна операція вигляду
 $\varphi_r : \Psi(R_1) \times \Psi(R_2) \xrightarrow{\sim} \Psi(R_1 \cup R_2)$, де
 $\text{dom } \varphi_r = \text{dom } \varphi$, причому

$$\varphi_r(\langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle) = \varphi(\langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle) \cup_{All}^{R_1 \cup R_2} \langle \psi_2 - \psi_1, R_2 \rangle \otimes_{R_2, R_1 \setminus R_2} \langle \{s_{R_1 \setminus R_2}^{NULL}\}, R_1 \setminus R_2 \rangle.$$

Під повним зовнішнім з'єднанням, індукованим операцією φ , розуміється часткова бінарна операція вигляду
 $\varphi_f : \Psi(R_1) \times \Psi(R_2) \xrightarrow{\sim} \Psi(R_1 \cup R_2)$, де
 $\text{dom } \varphi_f = \text{dom } \varphi$, причому

$$\varphi_f(\langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle) = \varphi(\langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle) \cup_{All}^{R_1 \cup R_2} \langle \psi_1 - \psi_2, R_1 \rangle \otimes_{R_1, R_2 \setminus R_1} \langle \{s_{R_2 \setminus R_1}^{NULL}\}, R_2 \setminus R_1 \rangle \cup_{All}^{R_1 \cup R_2} \langle \psi_2 - \psi_1, R_2 \rangle \otimes_{R_2, R_1 \setminus R_2} \langle \{s_{R_1 \setminus R_2}^{NULL}\}, R_1 \setminus R_2 \rangle.$$

Під зовнішнім з'єднанням об'єднанням розуміється часткова бінарна операція вигляду
 $\varphi_{\cup} : \Psi(R_1) \times \Psi(R_2) \xrightarrow{\sim} \Psi(R_1 \cup R_2)$, де
 $\text{dom } \varphi_{\cup} = \text{dom } \varphi$, причому

$$\varphi_{\cup}(\langle \psi_1, R_1 \rangle, \langle \psi_2, R_2 \rangle) = \langle \psi_1 - \psi_2, R_1 \rangle \otimes_{R_1, R_2 \setminus R_1} \langle \{s_{R_2 \setminus R_1}^{NULL}\}, R_2 \setminus R_1 \rangle \cup_{All}^{R_1 \cup R_2} \langle \psi_2 - \psi_1, R_2 \rangle \otimes_{R_2, R_1 \setminus R_2} \langle \{s_{R_1 \setminus R_2}^{NULL}\}, R_1 \setminus R_2 \rangle.$$

Висновки

У статті визначено формальну математичну семантику операцій внутрішніх і зовнішніх з'єднань та операції напівз'єднання над таблицями як мультимножинами рядків однієї схеми. Розглянуто такі операції внутрішнього з'єднання як декартове з'єднання, внутрішнє з'єднання за атрибутами $A_1, \dots, A_n$, внутрішнє природне з'єднання, внутрішнє з'єднання за предикатом.

Для задання операцій зовнішніх з'єднань введено особливий елемент універсального домену $NULL$. Розглянуто чотири операції зовнішнього з'єднання: зовнішнє ліве з'єднання, зовнішнє праве з'єднання, повне зовнішнє з'єднання, зовнішнє з'єднання об'єднання.

Список використаних джерел

1. Codd E.F. Relational Completeness of Data Base Sublanguages // Data Base Systems. – New York: Prentice-Hall. – 1972. – P. 65-93.
2. Grefen Paul W.P.J., Rolf A. De By A Multi-Set Extended Relational Algebra. A Formal Approach to a Practical Issue // 10th International Conference on Data Engineering, ICDE, February 14-18, 1994, Houston, TX, USA. – 1994. – P. 80-88.
3. Lamperti G., Melchiori M., Zanella M. On Multisets in Database Systems // Multiset Processing: Mathematical, Computer Science, and Molecular Computing Points of View, number 2235

in Lecture Notes in Computing Since. – Berlin: Springer-Verlag. – 2001. – P. 147-215.

4. Silbeschatz A., Korth H., Sudarshan S. Database System Concepts – McGraw-Hill, 2011. – 1376 p.

5. Redko, V., Brona, J., Buy, D., Poliakov, S.: Relation database: relation algebras and SQL-similar languages. – Kyiv. – 2001. (in Ukrainian)

6. Glushko I.M. Multiset table algebra // Proceedings of the International Scientific Conference of Student and Young Scientists "Theoretical and Applied Aspects of Cybernetics", February 21–25, 2011, Kyiv. – 2001. – P. 77-79. (in Ukrainian)

Надійшла до редколегії 26.06.2013

УДК 004.056.5

Гончар С.А.¹, аспірант

Групова схема Діффі-Хеллмана, яка базується на кучі

У даній статті представлено нову структуру відкритого обміну ключами у групі, а саме групову схему Діффі-Хеллмана, яка базується на кучі. Показано її ефективність в порівнянні з існуючими підходами.

Ключові слова: криптографія, дерево, куча, груповая схема Діффі-Хеллмана.

¹ Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т Глушкова 4д, e-mail: sg.gonchar@google.com

S. A. Gonchar ¹, postgraduate

Heap Diffie-Hellman Group

This paper presents new structure for public key exchange in group, namely the Heap Diffie-Hellman Group scheme (HDHG). Shown its effectiveness in comparison with existing approaches.

Key Words: cryptography, tree, heap, group Diffie-Hellman scheme.

¹ Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova av., 4d, e-mail: sg.gonchar@google.com

Статтю представив чл.-кор. НАНУ, д.ф.-м.н., проф. Анісімов А.В.

Вступ

Безпечна передача даних є зараз дуже актуальною проблемою в криптографії, оскільки в більшості випадків нам необхідно передати цю інформацію так, щоб її не перехопив зловмисник.

"Захищені групові комунікації" відносяться до сценаріїв, у яких група учасників може відправляти і отримувати повідомлення членів групи, таким чином, що сторонні особи не мають змоги отримати цю інформацію, навіть якщо вони перехоплюють повідомлення. Захищені групові комунікації ґрунтуються на протоколах та теоретичних інструментах сучасної криптографії, які швидко розвиваються.

На даний час в практиці широко застосовуються групові схеми Діффі-Хеллмана обміну ключами[1]. Ці схеми дозволяють генерувати криптографічні ключі у відкритих системах, обмінюючись між учасниками групи закритими ключами. Весь цей обмін закритими ключами є довготривалим, оскільки при обміні ключами кожному з учасників приходится робити великі обчислення модульного піднесення до степеня і лише після обрахувань проводити пересилку свого закритого ключа відповідним учасникам.

Оскільки у групі можуть перебувати учасники з різною обчислювальною можливістю, процес

генерування ключа у них може затягуватись на певний час, тим самим сповільнюючи процес отримання групового ключа. У криптографії кожна затримка грає велику роль, а тому основною задачею для процесу створення групового ключа є мінімізація цих затримок.

Існуючі підходи

У сучасній криптографії широкого застосування набув так званий *TGDH (Tree Group Diffie-Hellman)* – груповая схема Діффі-Хеллмана, що базується на дереві[2]. У даній схемі ключового обміну учасники групи виступають вершинами дерева, а зв'язки між ними – ребрами. Оскільки дана схема базується на дереві, це дає нам можливість виконувати паралельні обрахунки на кожній з вершин дерева (у кожного учасника), тим самим пришвидшуючи швидкість обміну даними та генерацію групового ключа. Особливо ефективно ця схема працює при великій кількості учасників у групі.

Існують також деякі модифікації цього підходу. Пропонується використовувати на кожному з рівнів дерева найбільш ефективного учасника для подальших обрахунків (*The Enhanced Tree-based Group Diffie-Hellman*)[3]. Це дало змогу пришвидшити основну схему ключового обміну.

Групова схема, що базується на дереві, наведена на малюнку 1.

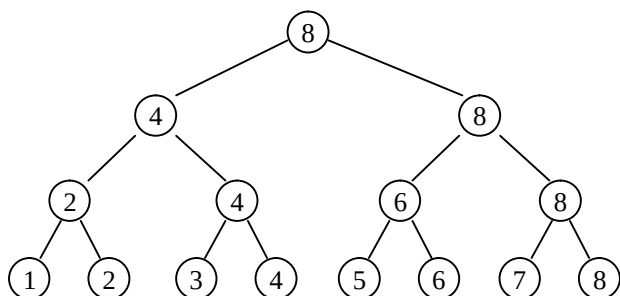


Рис. 1 Групова схема обміну ключами TGDH

Як було сказано вище, вона має багато переваг, оскільки дана схема дозволяє виконувати паралельні обчислення на кожному з етапів, але також у неї є й свої недоліки.

Як видно на малюнку 1, дерево у нас є збалансованим. Оскільки у нас група не є константною на протязі групової сесії (учасники можуть приєднуватись/вилучатись з групи), нам постійно потрібно підтримувати дане дерево збалансованим. А тому основним недоліком в цьому підході є те, що з кожною операцією приєднання/вилучення певного учасника до групи нам приходится балансувати дане дерево, оскільки в незбалансованому дереві швидкість обрахунків буде зростати, а це призводить до додаткових обрахунків.

Групова схема Діффі-Хеллмана, яка базується на кучі (HDHG).

Основою на групових схемах ключового обміну Діффі-Хеллмана, я хочу запропонувати інший підхід, який окрім усіх операцій, буде також обраховувати та використовувати у подальших обрахунках швидкодію усіх учасників у групі. Новизною у цьому підході є те, що ми не будемо використовувати дерево, для представлення учасників у групі, а будемо представляти їх в структурі даних, яка носить назву – куча[4].

Для обрахування швидкодії учасників я пропоную використовувати швидкість відклику певного учасника в процесі генерування ключа. Логічно зробити з цього припущення, що чим швидшим буде відклик учасника, тим швидшим буде його обчислювальна потужність та канал зв'язку, що в подібних обчисленнях грає велику роль.

Куча – це спеціалізована структура даних типу дерева, у якій повинна виконуватись наступна

властивість : якщо B являється вузлом-нащадком A , то $ключ(A) \geq ключ(B)$ ($ключ(A) \leq ключ(B)$). З цього випливатиме той факт, що кореневим вузлом кучі завжди буде учасник, у якого буде найбільший (найменший) ключ. Під ключем в даному випадку матиметься на увазі швидкість відклику конкретного учасника у групі.

В порівнянні з деревом, у якого структура даних повинна бути деревом, куча дозволяє працювати лише з масивом, який постійно буде відсортований. Для того, щоб даний масив мав зв'язки з іншими, у нас i -й елемент масиву матиме посилання на $2*i$ та $2*i+1$ елементи. В даному підході я пропоную відсортовувати його по тому, на скільки швидко кожен із учасників виконав обрахування (на скільки швидко прийшов відклик від певного учасника).

Також однією з переваг даного підходу є те, що використання кучі дозволяє нам динамічно змінювати позицію кожного учасника в залежності від того, як збільшиться або зменшиться швидкість його обрахунків (змінилась швидкість передачі даних по каналу зв'язку, комп'ютер учасника отримав інше навантаження, а тому швидкість обчислень у нього зменшилась, і т.д.). У випадку зміни швидкості обчислень нам прийдеється виконати операцію зміщення певного учасника в кучі (масиві), складність якої буде $O(\log n)$ (при використанні хеш-таблиці для збереження даних). Використання такої динамічної структури дасть нам змогу пришвидшити швидкість створення нового групового ключа, оскільки знаючи швидкодію кожного з учасників, ми будемо використовувати тільки найефективніших учасників на кожному з етапів даної схеми.

Для 6-ти учасників дана схема матиме наступний вигляд :

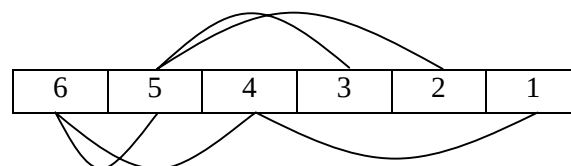


Рис. 2 Представлення кучі для 6-ти учасників у вигляді масиву

Вважатимемо, що учасники (1)-(6) мають швидкість відклику від 1 до 6 відповідно. Тому, як видно на малюнку 2, куча у нас буде мати вигляд відсортованого масиву по швидкодії відклику певного учасника. Дугами, на вище

наведеному малюнку, позначені відповідні зв'язки між учасниками, використовуючи які, можна легко побудувати дерево.

На малюнку 3 наведено дерево, яким можна представити даних 6 учасників, використовуючи зв'язки з малюнка 2.

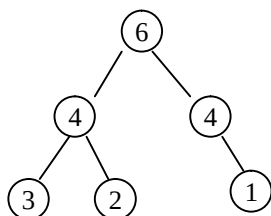


Рис. 3 Представлення кучі для 6-ти учасників у вигляді дерева

Оскільки куча представляється в основному у вигляді масиву, у якому задана схема по якій встановлюються зв'язки між елементами, у нас завжди отримане дерево буде збалансованим.

Розглянемо основні схеми приєднання/вилучення учасників з групи в момент існування сесії.

Схема приєднання нового учасника

При використанні кучі у нас не буде проблем, коли у нас відбуватиметься приєднання нового учасника до групи, у порівнянні з підходом, де ми використовуємо дерево. При приєднанні нового учасника до групи ми вставлятимемо його в масив туди, на скільки він буде продуктивним в порівнянні з іншими учасниками групи. Як говорилось уже вище, нам не прийдеється виконувати балансування в кучі, оскільки вона у нас задаватиметься масивом і при приєднанні нового учасника в виділене місце в масиві, у нас просто змінюватимуться посилання на них. Складність вставки нового елемента при використанні хеш-таблиці буде $O(\log n)$. При використанні підходу з деревом, у нас потрібно буде окрім вставки елемента, виконувати ще й балансування дерева. Використовуючи відомі алгоритми балансування дерева (АВЛ-дерево[5], 2-3 дерево[6]) у нас саме балансування відбуватиметься за час $O(\log n)$.

Покажемо наочно, як саме відбуватиметься вставка нового елемента в групі.

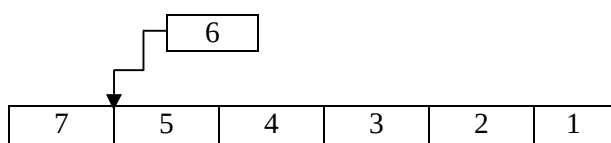


Рис. 4 Приєднання нового учасника

Приєднавши нового учасника в групу у нас в масиві зміняться відповідно і усі посилання на кожного з учасників, а тому структура масиву матиме наступний вигляд :

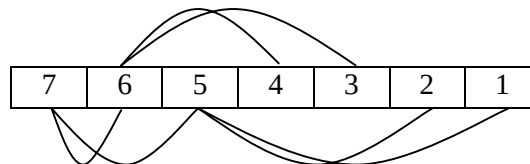


Рис. 5 Вигляд кучі після приєднання нового учасника.

Схема вилучення існуючого учасника

В свою чергу при вилученні існуючого учасника у нас просто зміститься сам масив, що при використанні хеш-таблиці матиме складність $O(\log n)$. І у нас знову не треба буде виконувати певні операції з балансуванням, чого не скажеш про використання дерева.

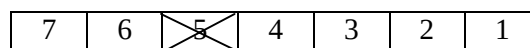
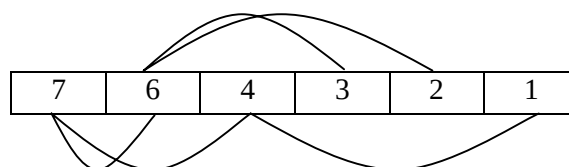


Рис. 6 Вилучення існуючого учасника

Після вилучення учасника наш масив матиме наступний вигляд :



Мал. 7 Вигляд кучі після вилучення існуючого учасника

Основні схеми обміну ключами, яка базується на кучі

Розглянувши підходи, які ґрунтуються на дереві, та запропонований підхід з використанням кучі, може виникнути певна неоднозначність, як саме використовувати кожного учасника у відповідних розрахунках на кожному з етапів створення групового ключа, а саме, хто із них повинен виконувати більше обчислень, хто менше і т.д.

Оскільки куча є структурою, у якій постійно усі елементи відсортовані, в даному випадку по швидкодії відклику, ми можемо виконувати обрахунки по різному. Я пропоную використовувати наступні 3 підходи в залежності від типу задачі.

1) *Ділення навпіл* – даний підхід дасть змогу використовувати на кожному етапі створення нового групового ключа найбільш ефективних учасників. Для знаходження найбільш ефективного учасника братиметься до уваги позиція учасника у кучі. А тому, виконуючи ділення навпіл у кучі, у нас малоефективні учасники взаємодітимуть з високоефективними учасниками враховуючи їх позицію, тобто маючи n учасників, операції модульного піднесення до степеня будуть проходити спочатку між 1-м і n -м учасником, 2-м і $n-1$ учасником, і т.д. Після виконання однієї з операцій, куча знову буде ділитись з поміж тих, хто брав участь в обрахунках на попередньому етапі і т.д., поки у нас не буде обраховано груповий ключ. Даний підхід може бути використаний тоді, коли у нас немає великих розбіжностей у швидкодії відклику учасників (приблизно однакові обчислювальні можливості).

Розглянемо приклад використання за участю 6-ти учасників :

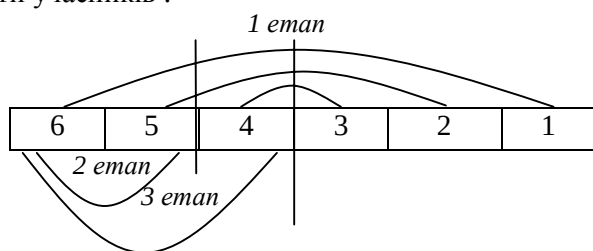


Рис. 8 Використання підходу ділення навпіл для 6-ти учасників

Оскільки кількість учасників парна, на 1-му етапі у нас куча поділиться на половину, і обрахунки у нас проходитимуть спочатку між (6) і (1), (5) і (2) та (4) і (3) учасниками, де основні обчислення лягатимуть на (6), (5) і (4) учасника відповідно. На 2-му етапі у нас уже непарна кількість учасників, а тому ділення виконуватиметься так, щоб обрахунки проходили між самими ефективними учасниками, а це (6) і (5). І нарешті на останньому етапі у нас залишиться виконати обчислення між (6) і (4) учасниками. Після виконання обчислень, буде встановлено груповий ключ.

2) *Встановлення порогу ефективності* – даний підхід може бути використаний, коли у нас в групі спостерігається різка зміна ефективності обрахунку у кожного з учасників групи. В такому випадку у нас в групі можуть бути комп'ютери, швидкість обрахунку у яких може перевищувати у декілька разів швидкість

обрахунку інших учасників. А тому в такому випадку доцільним є застосувати певний поріг, по якому ми будемо фільтрувати учасників у групі та застосовувати в обрахунках лише тих учасників, швидкість обрахунку яких вища або дорівнює заданому порогу. В даному випадку у нас уже група рідко буде ділитись навпіл, а тому коли більш ефективні учасники групи братимуть участь в обрахунках, отримавши закриті ключі від більш слабких учасників, інші більш слабкі учасники у групі, будуть очікувати, коли закінчиться обрахунок ключа для тих, хто уже відправив свої закриті ключі, і лише тоді відправлятимуть свої ключі для обрахунку. Цей процес відбуватиметься до того часу, поки не буде створено груповий ключ.

Розглянемо приклад використання за участю 6-ти учасників :

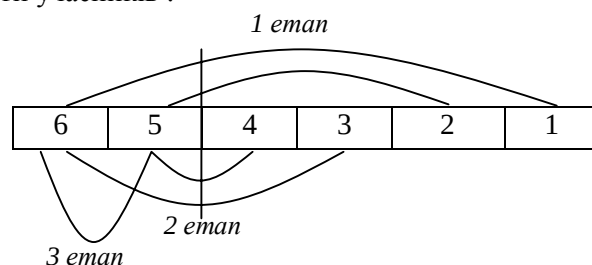


Рис. 9 Використання підходу з заданим порогом для 6-ти учасників

Нехай у нас ліміт буде задаватись швидкістю 5. Тоді відповідно участь в усіх обчисленнях у нас прийматимуть учасники (5) та (6), які знаходяться в рамках цього порогу.

В даних обчисленнях у нас на 1-му етапі відбуватимуться обрахунки між учасниками (6) і (1) та (5) і (2). Учасники (4) та (3) очікуватимуть, поки виконаються ці обчислення. На 2-му етапі відбуватимуться обрахунки між учасниками (6) і (3) та (5) і (4). Після завершення цих обрахунків, буде виконано обрахунок між (6) і (5) учасниками, після чого буде встановлено груповий ключ.

3) *Використання сервера (серверів) в обчисленнях* – даний підхід матиме велике застосування, якщо у нас в групі будуть присутні сервери, на які ми захочемо скинути практично усі обрахунки. Оскільки швидкість серверу буде найбільш ефективною, і сервер теж буде належати до нашої кучі (буде приймати участь в обрахунках), ми можемо відправляти закриті ключі від учасників до сервера, і уже він буде оперувати обчисленнями. Схема встановлення групового ключа частково буде нагадувати схему, наведену в 2-му підході. На відміну від 2-

го підходу, в якості учасників, які будуть приймати участь в обчисленнях, у нас будуть виступати сервер (сервери).

Усі зазначені підходи можуть мати ті чи інші застосування у певних сферах, де присутні одні з наведених параметрів. У порівнянні з підходом що базується на дереві, куча дозволяє по різному розподілити обчислення. В свою чергу дерево не може забезпечити такої ж гнучкості, оскільки в кожному піддереві повинен бути конкретний учасник, який буде виконувати обчислення. І він може проводити обчислення лише на певних рівнях, а тому використання одного і того ж учасника уже є неможливим в цьому підході.

Також, оскільки куча є уже відомою структурою даних, на даний час існує багато підходів, які дозволяють використовувати різні схеми для її побудови, кожна з яких має ті чи інші недоліки.

Нижче подана таблиця представлень кучі, які указують на швидкість роботи певних операцій в кучі. Різні представлення будуть давати різні оцінки складності.

Табл. 1

Схеми побудови кучі

Операція	Двійкова	Біноміальна	Фібоначчі	Спарена	Бродал
Знайти min(max)	$\Theta(1)$	$\Theta(\log n)$ або $\Theta(1)$	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$
Видалити	$\Theta(\log n)$	$\Theta(\log n)$	$O(\log n)^*$	$O(\log n)^*$	$O(\log n)$
Добавити	$\Theta(\log n)$	$O(\log n)$	$\Theta(1)$	$O(1)^*$	$\Theta(1)$
Зменшити ключ	$\Theta(\log n)$	$\Theta(\log n)$	$\Theta(1)^*$	$O(\log n)^*$	$\Theta(1)$
Злити	$\Theta(n)$	$O(\log n)^{**}$	$\Theta(1)$	$O(1)^*$	$\Theta(1)$

Приклад використання

Розглянемо детально принцип роботи запропонованого підходу з використанням групової схеми Діффі-Хеллмана для 8-ми учасників з використанням підходу ділення навпіл.

Схема обміну матиме наступний вигляд :

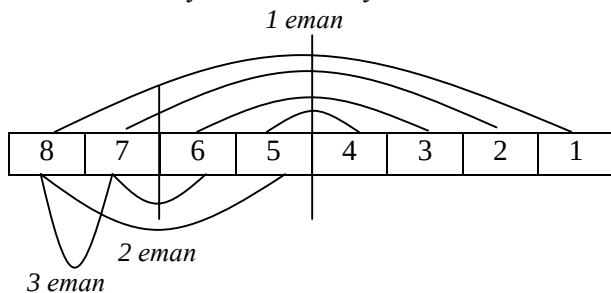


Рис. 10 Використання підходу ділення навпіл для 8-ми учасників

Розглянемо основні кроки створення групового ключа з використанням запропонованої схеми :

- Генерується 'g' для всієї групи;
- Кожний учасник $P_1, P_2, P_3, P_4, P_5, P_6, P_7$ і P_8 генерують для цього сеансу свої $n_1, n_2, n_3, n_4, n_5, n_6, n_7$ і n_8 відповідно;
- Виконуються обрахування $g^{n_1}, g^{n_2}, g^{n_3}, g^{n_4}, g^{n_5}, g^{n_6}, g^{n_7}$ і g^{n_8} відповідно у кожного з учасників;
- Виходячи з швидкості відклику кожного з учасників на попередньому етапі обрахувань, у нас створюється куча, яка буде відсортована по спаданні швидкодії учасників.
- Виконується 1-й етап обрахувань між учасниками :

Обрахування між (8) і (1) – (1)-й учасник виконує пересилку g^{n_1} до (8)-го учасника. (8)-й учасник виконує обрахування і отримує $g^{n_1 n_8}$;

Обрахування між (7) і (2) – (2)-й учасник виконує пересилку g^{n_2} до (7)-го учасника. (7)-й учасник виконує обрахування і отримує $g^{n_2 n_7}$;

Обрахування між (6) і (3) – (3)-й учасник виконує пересилку g^{n_3} до (6)-го учасника. (6)-й учасник виконує обрахування і отримує $g^{n_3 n_6}$;

Обрахування між (5) і (4) – (4)-й учасник виконує пересилку g^{n_4} до (5)-го учасника. (5)-й учасник виконує обрахування і отримує $g^{n_4 n_5}$;

- Виконується 2-й етап обрахувань між учасниками :

Обрахування між (8) і (5) – (5)-й учасник виконує пересилку $g^{n_4 n_5}$ до (8)-го учасника. (8)-й учасник виконує обрахування і отримує : $g^{n_1 n_8} g^{n_4 n_5}$;

Обрахування між (7) і (6) – (6)-й учасник виконує пересилку $g^{n_3 n_6}$ до (7)-го учасника. (7)-й учасник виконує обрахування і отримує : $g^{n_2 n_7} g^{n_3 n_6}$;

- Виконується 3-й етап обрахувань між учасниками :

Обрахування між (8) і (7) – (7)-й учасник виконує пересилку $g^{n_2 n_7} g^{n_3 n_6}$ до (8)-го учасника. (8)-й учасник виконує обрахування і отримує :

$g^{g^{n_1 n_8} g^{n_4 n_5} g^{n_2 n_7} g^{n_3 n_6}}$ - груповий ключ.

Усі піднесення до степеня, які виконувались на кожному кроці у вище описаній схемі, повинні виконуватись по модулю.

Висновки

Генерація групового ключа є важливою задачею в управлінні груповими ключами. Управління груповими ключами фокусується на зменшенні обчислювальних витрат, що є по суті самими важливими криптографічними операціями. Зрозуміло, що усі ключі, які генеруються для шифрування інформації у групі використовують великі числа, які підносяться до степеня по модулю, а тому для таких задач потрібні великі обчислювальні потужності зі сторони користувачів.

Оскільки сучасні обчислювальні машини дуже швидко розвиваються, з кожним роком проблема криптографії стає все більш актуальною, тому що для шифрування інформації приходиться генерувати все більші і більші ключі, а це складно, і тому перед нами постають питання використовувати уже існуючі алгоритми для генерування криптографічних ключів, чи знаходити нові методи в їх генерації, які будуть давати можливість працювати з меншими обчислювальними потужностями.

Запропонований підхід у даній статті, що базується на кучі, дозволяє використовувати обчислювальні потужності комп'ютерів учасників таким чином, що на кожному етапі обрахунку ключів у нас прийматимуть участь в процесі обрахунку найбільш ефективні учасники. А це в свою чергу дасть змогу пришвидшити швидкість створення нового ключа у групі.

Запропоновані вище підходи можуть бути широко застосовані на практиці.

Підходи ділення навіпіл та з заданим лімітом можуть ефективно використовуватись в групових комунікаціях, телеконференціях і т.д., де участь в груповій сесії будуть приймати лише користувачі і немає необхідності використовувати сервер в цій сесії.

Підхід, який базується на використанні сервера (серверів) буде дуже ефективним, якщо його застосувати до мобільних телефонів або іншої техніки, де обчислювальні можливості набагато менші, ніж обчислювальні можливості сучасних комп'ютерів. Використання даного підходу в мобільних телефонах дасть змогу робити громіздкі обрахунки на серверах, і лише невеликі обрахунки уже безпосередньо на телефонах, а це матиме великий приріст в їх роботі.

Список використаних джерел

1. Bresson E., Chevassut O., Pointcheval D., Quisquater J., Provably Authenticated Group Diffie-Hellman Key Exchange. – Philadelphia : ACM Press, Computer and Communication Security. – 2001. – November 5-8. – P. 255-264.
2. Kim Y., Perrig A., Tsudik G., Tree-based Group Key Agreement. – ACM Transactions on Information and System Security. – 2004.
3. Hong S., Secure and Efficient Tree-based Group Diffie-Hellman Protocol. – KSII Transactions and Information Systems. – 2009. – VOL. 3. – NO 2.
4. Heap (data structure). - [http://en.wikipedia.org/wiki/Heap_\(data_structure\)](http://en.wikipedia.org/wiki/Heap_(data_structure)).
5. AVL Tree. - http://en.wikipedia.org/wiki/AVL_tree.
6. 2-3 tree. - http://en.wikipedia.org/wiki/2%E2%80%933_tree.

Надійшла до редколегії 10.09.13

УДК 681.3

Олександр В. Дерев'янченко, к.ф.-м.н., доц.

Сучасні підходи до автоматизованого тестування користувацького інтерфейсу під ОС Android

В статті проаналізовані існуючі підходи до автоматизації тестування користувацького інтерфейсу загалом, розглянуто структуру та основні компоненти системи Android, а також описаний існуючий підхід до автоматизованого тестування графічного інтерфейсу та шляхи покращення тестування в ній.

Ключові слова: автоматизоване тестування, тестування інтерфейсу користувача, фреймворк для тестування, Android.

Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Глушкова 4д, e-mail: alexanderder@mail.ru

Oleksandr V. Derevyanchenko, Ph.D. (Physics & Mathematics), Associate Professor

Existing approaches to automated user interface testing of the Android system

This article analyzes existing approaches to automated user interface testing in general, a structure and main components of the Android system, and also describes basic existing nowadays approach to automated GUI testing and the ways to improve testing in this system.

Key words: automated testing, UI testing, testing framework, Android.

Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova st., 4d, e-mail: alexanderder@mail.ru

Статтю представив чл.-кор. НАН України, д.ф.-м.н., проф. Анісімов А.В.

Ця робота присвячена питанням автоматизованого тестування [1] користувацького інтерфейсу, а саме швидкої розробки автоматизованих тестів під платформу Android [2].

Даний напрямок автоматизації тестування графічного інтерфейсу під Android [3-5], дуже актуальний на сьогодні. Розробка програмного забезпечення під мобільні пристрої є дуже швидким процесом, тому дуже гостро постає питання тестування графічного інтерфейсу розроблюваних додатків, особливо в разі впровадження нового функціоналу. Тому написання тестів повинно відбуватись швидко, надійно та на якісно високому рівні з метою подальшого підтримання програмного коду тестів.

Загальний підхід до тестування інтерфейсу користувача (UI) це провести тести вручну і переконавшись, що додаток працює так як від нього очікується. Тим не менш, цей підхід може бути трудомістким, стомлюючим і є більш схильним до генерації помилок. Більш ефективний і надійний

підхід полягає в автоматизації тестування користувацького інтерфейсу в рамках тестування програмного забезпечення. Автоматизоване тестування включає в себе створення програм для виконання тестових завдань (тестів) для покриття конкретних сценаріїв використання, а потім за допомогою тестувальних засобів (або фреймворків) для запуску тестів автоматично перевірка, що все було відтворено та виконано в правильному порядку та правильним чином.

Сучасні підходи використання ОС Android [4-5] вимагають написання тестів з використанням інструментарію *UIAutomator*, що є більш низьким рівнем розробки тестів.

Тобто при ручному заданні елементів графічного інтерфейсу, без необхідного рівня абстракції від конкретної реалізації. Адже розробка програмного продукту під платформу Android характеризується великою кількістю різних розмірів екранів мобільних пристроїв, що робить ручний спосіб написання тестів надто громіздким.

В роботі пропонується концепція фреймворку автоматизованого тестування, що мотивована відсутністю ефективного за швидкістю механізму створення тестів графічного інтерфейсу. Можливість швидкої побудови об'єктної моделі для довільних елементів графічного інтерфейсу має вирішальне значення, для програмістів, що створюють автоматизовані тести.

В роботі розглядається стандартна методика UI-тестування під ОС Android та пропонується концепція розробки фреймворку, що дозволить створити певний рівень абстракції між існуючим інструментом *UIAutomator* та графічним інтерфейсом програмного додатку під ОС Android. Система має представляти графічні елементи як певні об'єкти та надавати тестувальнику можливість швидко та якісно проводити тести і мати можливість легко їх змінювати.

На даний час в поставці Android SDK існують наступні засоби для автоматизованого тестування програмного забезпечення написаного з використанням користувацького інтерфейсу:

UiAutomatorViewer – інструмент, що дозволяє сканувати і аналізувати елементи графічного інтерфейсу програмного додатку під Android. Працює на базі прошарку API в системі, що відповідає за доступність кожного UI елементу (accessibility).

UIAutomator – бібліотека написана мовою Java для створення кастомізованих функціональних UI тестів, що можуть бути виконані за допомогою запускателя тестів вбудованого в платформу Android [4].

Для використання даних засобів потрібно задовольнити наступні вимоги: Android ADK Tools ревізії 21 або вище, а також Android SDK API 16 або вище.

Послідовність виконання дій при UI-тестуванні під ОС Android

Наведемо методик у послідовності виконання дій при автоматизованому тестуванні графічного користувацького інтерфейсу [4]:

- Підготовка до тестування. До неї входить встановлення тестованого програмного додатку на пристрій, аналіз графічного інтерфейсу додатку, а також перевірка того, що графічні елементи додатку доступні для тестування.
- Створення автоматизованих тестів для симуляції специфічних користувацьких дій над програмним додатком.
- Компіляція тестових сценаріїв у JAR-архів, що виконується та інсталяція його на пристрій разом з тестованим додатком.
- Запуск автоматизованого тестування та перегляд отриманих результатів.
- Виправлення та корекція помилок та програмних дефектів виявлених під час тестування.

Перед тим, як приступити до написання тестів, необхідно ознайомитися з компонентами користувацького інтерфейсу (у тому числі бути знайомим з видами елементів представлення графічного інтерфейсу в системі Android) тестованого програмного додатку. Для цього і стане в пригоді інструмент для візуалізації графічного інтерфейсу з точки зору програмного представлення. *UIAutomatorviewer* робить знімок екрана користувацького інтерфейсу на будь-якому пристрої Android, підключеному до комп'ютеру на якому запущений даний інструмент. *UIAutomatorviewer* забезпечує зручний візуальний інтерфейс для перевірки макета ієрархії візуальних елементів UI і перегляду властивостей окремих компонентів користувацького інтерфейсу, які відображаються на тестованому пристрої. Використовуючи цю інформацію, можна пізніше створити *uiautomator*-тести з об'єктами селектор, які орієнтовані на конкретні компоненти для користувацького інтерфейсу для тестування.

Перед тим як використовувати тестувальний інструмент *UIAutomator* потрібно виконати наступні етапи:

- Завантажити програмний додаток на пристрій. Переконайтесь, що тестований додаток встановлений на пристрої (що

- версія саме та, яку передбачається тестувати).
- Дослідити користувацький інтерфейс тестованої програми. Використовуючи раніше описаний *UIAutomatorviewer* для отримання представлення ієрархії макетів та графічних елементів. Найчастіше всього елементи з якими взаємодіє людина є елементами з певними візуальними характеристиками, такими як текст, або специфічне графічне зображення, яке легко можна буде розрізнити використовуючи селекторні об'єкти тестувального інструменту. Якщо є доступ до вихідного коду тестованого додатку, то досить корисним може бути додавання до всіх графічних елементів опису (а саме поля контент-опису) для полегшення його доступності при автоматизованому тестуванні.
 - Зробити налаштування середовища розробки. Адже навіть при готовності всіх попередніх кроків, відсутність налаштованого середовища може призвести до затримок.

Створення автоматизованих тестів на *UIAutomator* інструменті

API тестового інструменту *UIAutomator* представлений наступними класами [4]:

- *UiDevice* – представляє собою стан мобільного пристрою що надає уніфікований доступ до різних його характеристик. А також надає легкий інтерфейс для виконання дій над графічним інтерфесом, такими як натискання кнопок, дотик до екрану та ін.
- *UiSelector* – представляє собою критерій пошуку графічних елементів, що представлені в певний момент часу на екрані пристрою. Якщо на екрані знаходиться декілька елементів, що підходять під пошуковий критерій, то повертається перший з них у ієрархії графічних елементів.

- *UiObject* – представляє собою об'єкт - елемент графічного інтерфейсу. Створюється на базі *UiSelector* об'єкту.
- *UiCollection* – наслідник класу *UiObject*. Представляє собою колекцію елементів графічного інтерфейсу.
- *UiScrollable* – наслідник класу *UiCollection*. Представляє собою колекцію елементів графічного інтерфейсу, що може бути проскролена.

Концепція фреймворку для покращення тестування під ОС Android

1) *Активіті*. Дуже часто при написанні тестів програміст зосереджується лише на конкретному моменті часу, та на певній множині дій. Не думаючи про майбутнє можливе розширення функціоналу програми. Що призводить до подальших проблем при впровадженні нового функціоналу в програму, адже тоді доводиться майже повністю переписувати тести.

В першу чергу графічний інтерфейс програм на платформі Android [4] це активіті. Таким чином легко передбачити, що першою основною компонентою нашого фреймворку будуть саме абстрактні об'єкти, що представляють собою активіті. А саме об'єкти, які будуть являти собою візуальне представлення для користувача. В свою чергу такі об'єкти мають свої параметри та набір методів. А також надають легкий та зручний інтерфейс доступу до вкладених у активіті графічних елементи. Такий підхід в першу чергу дозволяє легко будувати структуру графічних елементів, а також застосувати певний алгоритм локації їх у рамках обраної абстракції. Ми можемо легко розділити логіку тестів та конкретну специфікацію активіті, що тестується, в рамках цілого додатку.

На цьому ж рівні абстракції ми описуємо всі елементи активіті. Тут можна додати і характерні абстрактні класи елементів, інтерфейси доступу до них. Тобто створити абстракцію для елементів керування користувацьким інтерфейсом.

2) *Атомарні дії*. Мета даного рівня абстракції організувати дії над елементами

активіті, обгорнути їх в атомарні кроки. Де під поняттям крок ми розуміємо найменшу логічну дію. Розмір цих атомарних дій залежатиме вже від конкретної області програмного додатку та побудови тестів. Ці ж найменші атомарні дії і будуть компонентами для створення складених дій.

Можна виділити два основних класи атомарних дій: звичайна дія та дія з верифікацією. До перших можна віднести всі дії, що можуть бути виконані користувачем, а також абстрактні дії, які є лише загальними для своїх наслідників. До класу ж дій з верифікацією всі такі, що крім самого виконання дії ще і верифікують певні речі, такі як результат виконання дії або проміжні результати.

3) Складені дії. Складені дії будуються з кількох частин. А саме: підготовчих дій, основних кроків тесту з верифікацією і кроків, що закінчують. В цілому задача складених дій полягає у групуванні атомарних дій за сутностями для повторного їх використання. Це надаватиме змогу розробляти тести притримуючись основних правил написання об'єктно-орієнтованого коду.

Основою для складених дій є підготовчі кроки. Адже дані операції потрібно якомога більше оптимізувати з питань повторного використання коду для споріднених сутностей.

Після підготовчих кроків йдуть основні кроки. Вони повинні формуватись на більш високому рівні абстракції, де саме і будуть створюватись сценарії виконання тестів. Але чим краще буде розроблена гнучка система завдання атомарних та складених дій, тим легше і швидше можна буде створювати тести.

Останньою складовою є закінчуючі кроки. Вони є не менш важливими, адже після них ми маємо отримати певний результат (після верифікації), а також при можливості повернути стан системи до такого, щоб ми могли успішно продовжити тестування. Тобто не створювати побічні ефекти після виконання даного тесту.

Висновки

Таким чином, дослідження показали значні перспективи розробки механізмів автоматизованого тестування під ОС Android. Запропонована концепція фреймворку дозволить набагато швидше розробляти автоматизовані тести, використовуючи додатковий рівень абстракції, що дозволить користувачам або програмістам встановлювати зв'язок об'єктів з елементами графічного інтерфейсу. Для пошуку документальних даних про елементи графічного інтерфейсу тестувальник може попередньо створити об'єктно-орієнтовану модель графічного інтерфейсу, після чого швидко збільшувати кількість автоматизованих тестів використовуючи її.

У подальшому можливо вдосконалювати запропоновану систему в різних напрямках. Можна розробляти додаткові рівні абстракції характерні для певних класів програмного забезпечення під Android. За рахунок будови фреймворку, створені нові рівні абстракції можуть бути легко впровадженні та адаптовані. Також можна розробляти конкретні реалізації певних класів атомарних та складених дій з метою ще більшого пришвидшення написання автоматизованих тестів.

Список використаних джерел

1. Fewster M., Graham D. *Software Test Automation*. - ACM Press, 1999, 600p.
2. Komatinemi S., MacLean D., Hashimi S.Y. *Pro Android 3*. - Apress, 2011, 1200p.
3. Tsung-Hsiang Chang, Tom Yeh, Robert C. Miller. *GUI Testing Using Computer Vision*. - ACM, CHI, 2010, pp.1535-1544.
4. http://developer.android.com/tools/testing/testing_ui.html - UI Testing for Android.
5. http://www.vogella.com/articles/AndroidTesting/article.html#roboelectric_overview - Android application testing with the Android test framework.

Надійшла до редколегії 07.06.2013

УДК 519.9

Дорошенко А.Ю.¹, д.ф.-м.н., проф.,
Єгоров В.І.², аспірант В.І.,
Суслова Л.М.³, аспірант Л.М.

Фреймворк для автоматизованого створення веб-додатків

*У данній статті представлено фреймворк для
PHP – SamsonPHP, та створений завдяки ньому
абсолютно новий шаблон проектування веб-
застосувань - Web module-view-controller(WMC).*

Ключові слова: php, php framework, MVC,
WMVC, модульність.

¹Інститут програмних систем Національної
академії наук України, 03187, м. Київ, пр-т.
Академіка Глушкова, 40, корпус 5,
e-mail: doroshenkoanatoliy2@gmail.com

²Інститут програмних систем Національної
академії наук України, 03187, м. Київ, пр-т.
Академіка Глушкова, 40, корпус 5,
e-mail: egorov@samsonos.com

³Інститут програмних систем Національної
академії наук України, 03187, м. Київ, пр-т.
Академіка Глушкова, 40, корпус 5,
e-mail: lidiya.suslova@gmail.com

Doroshenko A.U.¹, professor,
Igorov V. I.² postgraduate,
Suslova L.M.³, postgraduate.

Framework for automation creation of web applications

*The paper introduces a new php framework named
SamsonPHP that introduces absolutely new design
pattern for web application – Web module-view-
controller(WMC).*

Key Words: php, MVC, WMVC, php framework,
modularity.

¹ Institute of Software Systems, The National
Academy of Sciences of Ukraine, 03187,
Kyiv, Glushkova st., 40, housing 5, e-
mail: doroshenkoanatoliy2@gmail.com

² Institute of Software Systems, The National
Academy of Sciences of Ukraine, 03187,
Kyiv, Glushkova st., 40, housing 5,
egorov@samsonos.com

³ Institute of Software Systems, The National
Academy of Sciences of Ukraine, 03187,
Kyiv, Glushkova st., 40, housing 5,
lidiya.suslova@gmail.com

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

Останнім часом web-додатки здобувають усе
більше популярності. З розвитком інтернету усе
гостріше постає проблема розробки інтернет
додатків з легко масштабованим і змінюваним
кодом. Головну роль у можливості такої
масштабованості відіграє архітектура проекту.
Найчастіше при розробці інтернет додатків
використовується архітектура Модель – Вид –
Контролер (Model-view-controller, MVC). Це
схема використання декількох шаблонів
проектування, за допомогою яких модель даних
додатка, користувацький інтерфейс і взаємодія з
користувачем розділені на три окремі частини
[1]. Модель містить об'єктну модель, бізнес-
логіку й доступ до даних, вид містить візуальний
спосіб відображення даних, генерацію сторінок,
а контролер — маршрутизацію запитів. Таким

чином, основна мета застосування цієї концепції
полягає в відокремленні бізнес логіки від її
візуалізації. Також дана схема проектування
часто використовується для побудови
архітектурного каркаса, коли переходять від
теорії до реалізації в конкретній предметній
області.

Ряд розроблювачів спеціалізується тільки в
одній з областей: або розробляють графічний
інтерфейс, або розробляють бізнес-логіку. Таким
чином, представляється можливим досягти того,
що програмісти, що займаються розробкою
бізнес-логіки (моделі), взагалі не будуть
інформовані про те, яке відображення буде
використовуватися.

Вид і контролер звичайно підтримуються
різними фреймворками й бібліотеками [2], [3],

[4]. Тоді як розробка моделі лягає на розроблювача.

Метою дослідження було вивчення різних середовищ розробки та створення своєї власної, яка б дозволила суттєво полегшити життя розроблювача веб-додатків за допомогою автоматизації безлічі рутинних процесів, надання можливостей мінімізації коду. Прагнучи створити середовище, що відповідало б поставленим вимогам, був розроблений фреймворк Samsonphp і шаблон WMVC – WEB Model-View-Controller pattern, який успадковує ідеї стандартного паттерна MVC і враховує особливості розробки в середовищі web.

За рахунок такого поділу підвищується можливість повторного використання коду. Найбільш корисне застосування даної концепції в тих випадках, коли користувач повинен бачити ті ж самі дані одночасно в різних контекстах і/або з різних точок зору.

Завдяки поділу, виконуються наступні завдання. До однієї моделі можна приєднати кілька видів, не зачіпаючи при цьому реалізацію моделі. Наприклад, деякі дані можуть бути одночасно представлені у вигляді електронної таблиці, гістограми та кругової діаграми. Не зачіпаючи реалізацію видів, можна змінити реакції на дії користувача (натискання мишею на кнопки, введення даних), для цього досить використовувати інший контролер. Також підвищується продуктивність кінцевих додатків завдяки використанню методу автоматичної генерації додатка в один скрипт [5].

Samson PHP

Samsonphp — це фреймворк, написаний мовою програмування PHP і призначений для швидкого й ефективного створення веб-сайтів і веб-додатків, з використанням PHP. Веб-сайти створювані з використанням Samsonphp повинні реалізовувати шаблон програмування WMVC(Model-view-controller/ Модель-Вистава-Контролер).

Також Samsonphp використовує принцип convention over configuration, який пропонує використання заздалегідь установлених правил і налаштувань, а не повноцінне конфігурування під кожний створюваний веб-сайт, але розроблювач може перевизначити кожне значення параметра фреймворка для власних потреб.

В основі розробки фреймворка лежить принцип програмування DRY (don't repeat yourself / не повторюй себе), який кардинально змінив підхід до створення веб-сайтів і веб-

додатків, і сильно відбився на структурі самого фреймворка.

Модуль — як основна частина веб-додатка

В Samsonphp два файли моделі, контролера й необмежена кількість файлів вигляду, що відносяться до однієї сутності, прийнято називати — Модуль. По факту кінцевий веб-сайт є набором модулів, за вивід яких відповідає ядро фреймворка Samsonphp.

Основна проблема існуючих фреймворків — відсутність повноцінної модульності при розробці веб-сайту. Внутрішня організація веб-серверів у наш час дозволяє доступ лише до файлів, які перебувають усередині локальної директорії веб-сайту (document root), тим самим обмежуючи нас від використання зовнішніх модулів. Тому всі сучасні PHP фреймворки використовують підхід до збирання модулів під конкретний веб-сайт або під групу однотипних веб-сайтів з наступним фізичним копіюванням цих файлів модулів у локальну папку кожного конкретного розроблювального веб-сайту. Такий підхід до файлової структури веб-сайту створює серйозні обмеження, труднощі, а в деяких випадках навіть помилки для програміста, тому що доводиться вручну або напівавтоматично підключати ці модулі до веб-сайту, контролювати їх версію і у випадку виходу оновлення модуля, копіювати його в кожний веб-сайт, який його використовує.

Мова програмування PHP дозволяє на серверній стороні підключати файли, які фізично перебувають поза файловою структурою веб-сайту, а конкретно поза його локальною директорією.

У термінології Samsonphp модуль — це не тільки набір з контролера, моделі та файлів виду, а також будь-яких інших ресурсів, таких як Javascript, CSS, картинка й будь-які інші файли. По суті, кожний модуль Samsonphp є міні веб-сайтом і може існувати окремо, якщо він для цього досить підготовлений, а саме: маючи свій контролер і шаблон виду. Цей новий підхід до розуміння модульності в розробці веб-додатків, дає нам повне право створювати окремі загальновикористовувані частини веб-сайту, виділяючи їх у повноцінні модулі будь-якої складності.

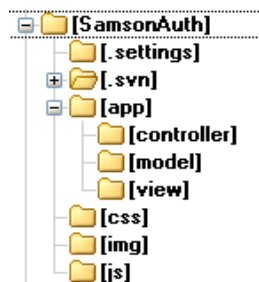


Рис. 1. Модуль Samsonauth

Такий підхід дозволяє створювати бібліотеки веб-ресурсів з яких кінцевий розроблювач зможе зібрати готовий повнофункціональний веб-сайт, змінивши лише при потребі файли виду та контролера цих модулів.

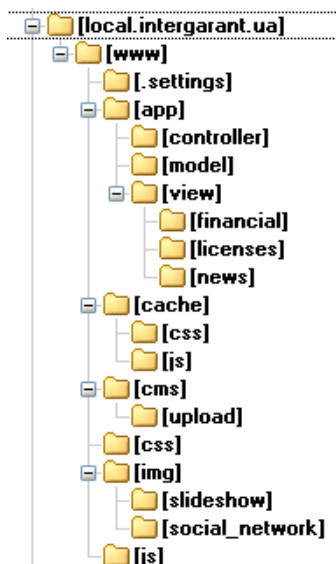


Рис. 2. Приклад сайту, реалізованого в Samsonphp

Основним недоліком при фізичному відокремленні файлів клієнтських ресурсів Javascript, CSS, картинок є абсолютна неможливість їх одержання клієнтським веб-браузером прямо від сервера, тому що веб-сервер не дає можливість звертатися до ресурсів веб-сайту поза його локальною директорією. Для надання доступу до таких ресурсів у зовнішніх модулях був розроблений спеціальний механізм маршрутизації Resourcerouter — роутер ресурсів.

Resourcerouter

Resourcerouter (роутер ресурсів) — це зовнішній модуль Samsonphp, який реалізований у вигляді системного контролера, який обробляє URL запити виду

`/src?a=@application&m=@module&r=@path`

Комбінація `@application` і `@module` називається маршрутом до джерела ресурсу. Для швидкого визначення потрібного маршруту з таблиці маршрутизації, у неї доданий спеціальний хеш ключ (KEY) з нього.

Механізм маршрутизації має наступний алгоритм, існує набір додатків (application), до яких ставляться модулі (module), усередині яких фізично зберігаються ресурси. Основною частиною Resourcerouter є таблиця маршрутизації, у якій зберігаються фізичне розміщення цих сутностей у файлової системи. Таблиця заповнюється та інтерпретується за наступним принципом:

- * означає будь-яке значення параметра в маршруті
- угорі таблиці розміщуються маршрути до додатка веб-сайту
- далі описуються маршрути до внутрішніх модулів веб-сайту

Таблиця 1

Маршрутизація ресурсів

Key	App	Module	Path
f3df58ef2 0f88671c 61da39e4 c008c15	*	system	E:\PHPPROJECTS \Samsonphp/
2791548c 939e9928 b8b829a3 3cb223c4	local	about	about/
e9986a0a 7dc8314d 4055c7c1 3cc57494	local	blog	blog/
2919674d 7c568d59 67dbb9d5 11bcb3cd	*	auth	E:\PHPPROJECTS \Samsonauth/
d861768d 27f08flab 398f29b9 baa7611	local	*	

Тепер для завершення необхідно всі шляхи до ресурсів усередині виглядів зовнішніх модулів і додатків замінити на спеціальну скорочену функцію фреймворка

`src( $path, $module = current )`

яка автоматично визначить із якого модуля й додатка вона викликається та сформує правильний URL маршрут до ресурсу:

`/src/?r=img/ajax_loader_1.gif&a=*&m=auth.`

Якщо один з параметрів маршруту упущений то він автоматично прирівнюється до “*”.

Приклад розбору URL маршруту:

1. Зазначений додаток — *, це означає що цей маршрут не ставиться до конкретного додатка, а значить значимою частиною є тільки модуль(auth)
2. Знаходимо рядок з Application — “*”, Module — “auth”.
3. Одержуємо, що фізично ресурси перебувають по шляху «E:\PHPPROJECTS\Samsonauth\»
4. Додаємо фізичному шляху відносно розміщення ресурсу усередині модуля «img/ajax_loader_1.gif»
5. Одержуємо шлях до ресурсу: «E:\PHPPROJECTS\Samsonauth\img/ajax_loader_1.gif»
6. Контролер зчитує вміст певного ресурсу, якщо він фізично існує й повертає його, указавши потрібний HTTP заголовок mime-type для браузера, у випадку його відсутності вертається заголовок 404 помилки

Таким чином, повністю вирішується завдання про одержання клієнтського доступу до ресурсів зовнішніх модулів веб-сайту.

Список використаних джерел

1. *Poskonin A.* Web-applications and data: problems of abstraction and scalability. – ISP RAN proceedings, 2012. – 159 p. (in Russian).
2. «Zend Framework», [In Internet]. URL: <http://framework.zend.com/>
3. «Code igniter», [In Internet]. URL: <http://ellislab.com/codeigniter>
4. «Slim PHP», [In Internet]. URL: <http://www.slimframework.com/>
5. *Botygin I.A.* Research of methods for increasing the productivity of web-applications. – Bulletin TPU. – 2008. – №5. – 109 p. (in Russian).

Phpcompressor

Phpcompressor (стискач PHP) — це зовнішній модуль Samsonphp, який служить для перетворення веб-сайту із середовища розробки у фінальний продукт, готовий до завантаження на веб-сервер. Таке перетворення необхідне для того, щоб не копіювати всю файлову структуру яка використовується при розробці веб-сайту, а так само для того, щоб веб-сайт функціонував на порядок швидше ніж його розподілена версія.

В Samsonphp процес перекладу веб-сайту зі стадії розробки у фінальну робочу стадію називають «стиском».

Даний модуль виконує лексичні перетворення й виконує фактичне збирання всього необхідного для функціонування веб-сайту в один фінальний пусковий файл index.php. Усі перетворення та операції копіювання, створення, переіменування ресурсів виконуються автоматично. Також спеціальний текстовий процесор, вбудований у модуль, має можливість забирати й змінювати PHP код модулів, які мають у собі спеціальні директиви для оптимізації роботи цих фрагментів коду в стислій версії сайту.

6. *Lantratov O.I., Samodelov A.N., Chumakiv S.E.* Experience of the module address to access the content of news pages of SDO GOU VPO «URGUES.» Portal. – Recent developments in science and education. – 2010. – №6. (addition of the «Engineering»). – 5 p. (in Russian).

7. *Tuzovskiy A.F.* Architecture of Semantic Web-Portal. – Bulletin TPU. – 2006. – №7. – 142 p. (in Russian).

Надійшла до редколегії 20.06.13

УДК 51.681.3

Кривий С.Л.¹ д. ф.-м. н., професор,
Максимець О.М.²

Аналіз властивостей мереж Петрі

У даній роботі розглядаються методи орієнтовані на верифікацію реактивних і розподілених систем за допомогою мереж Петрі. У ній представлений технологічний процес дослідження властивостей Мережі Петрі з використанням TSS-методу.

Ключові слова: формальна верифікація, реактивні системи, мережі Петрі.

¹ Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Глушкова 4д,
e-mail sl.krivoi@gmail.com

² Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Глушкова 4д,
e-mail maksymets@gmail.com

S.L. Kryvyi¹, Ph.D., Professor,
O.M. Maksymets²

Properties analysis of Petri nets

This paper reviews the methods focused on the verification of reactive and distributed systems with Petri nets. It presents a technological process of research of properties Petri nets using TSS-method.

Keywords: formal verification, reactive systems, Petri nets.

¹ Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova st., 4d,
e-mail sl.krivoi@gmail.com

² Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova st., 4d,
e-mail maksymets@gmail.com

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

Метою дослідження є побудова технологічного процесу аналізу властивостей мереж Петрі і використання розроблених для цього спеціальних алгоритмів. Актуальні дослідження в цьому напрямку, використовують загальні алгоритми лінійної алгебри в чому і полягає їх неефективність.

Визначення 1. Мережею Петрі (МП) називається трійка (P, T, F) , де P і T множини, які не перетинаються і елементи яких називаються місцями і переходами відповідно, а $F \subseteq (P \times T) \cup (T \times P)$ – відношення інцидентності. Елементи з F називаються стрілками, а місця – вершинами, маючи на увазі графічне представлення мережі. Якщо $(x, y) \in F$, то x називається вхідною вершиною y , а y – вихідною вершиною x . Множина вхідних і вихідних вершин для x позначається $\bullet x$ і $x \bullet$ відповідно.

Мережа (P, T, F) називається розміченою, якщо задана функція розміток $M: P \rightarrow N$, де N – множина натуральних чисел. Якщо $M(p) = m$, то це означає, що функція M ставить у вершину p рівно m фішок. Розмітка мережі, у

разі коли $|P| = n$ і множина P впорядкована, представляється вектором $M = (m_1, m_2, \dots, m_n)$, де $m_i = M(p_i)$, $p_i \in P$, $i = 1, 2, \dots, n$.

Мережею Петрі називається четвірка (P, T, F, W, M_0) , де (P, T, F) – мережа, M_0 – початкова розмітка її місць, а $W: F \rightarrow N \setminus \{0\}$ – функція кратності дуг МП [2].

Класи МП і їх властивості. Розглянемо класифікацію МП і методи дослідження їх найважливіших властивостей. З цією метою введемо деякі визначення загального характеру.

Визначення 2. Нехай для деякої МП (P, T, F, W, M_0) є підмножини $Q \subseteq P$ і $Q \neq \emptyset$. Множину Q називають

а) дедлоком (сифоном) тоді і тільки тоді, коли $\bullet Q \subseteq Q \bullet$;

б) пасткою тоді і тільки тоді, коли $Q \bullet \subseteq \bullet Q$.

Для МП (S, W, M_0) з n переходами і m місцями матрицею інцидентності A називається матриця з цілочисельними коефіцієнтами розмірності $n \times m$, елементи якої визначаються таки рівняннями $a_{ij} = I(t_j, p_i) - I(p_i, t_j)$, де I – відношення інцидентності даної МП.

Справедливість цього рівняння впливає з правила збереження фішок, оскільки коефіцієнт a_{ij} матриці інцидентності представляє число фішок, які переміщуються, змінюються і додаються в місце p_i при спрацюванні переходу t_j в МП.

Оскільки j -й рядок в матриці інцидентності A означає зміну розмітки в результаті спрацювання переходу t_j , то можна записати таке рівняння, яке називається рівнянням стану МП: $M_k = M_0 + A u_k$, де $k=1,2,\dots$, а A – матриця інцидентності.

Припустимо, що деяка фінальна розмітка M_k досяжна з початкової розмітки M_0 за допомогою послідовності переходів $t_1, t_2, \dots, t_k$. Записавши рівняння стану для всіх $j=1,2,\dots,k$ і додавши їх, отримуємо $M_k = M_0 + A \cdot \sum_{j=1}^k u_j$, або $Ax = d$, де $d = M_k - M_0$, а $x = \sum_{j=1}^k u_j$.

Система рівнянь $A \cdot x = d$ називається необхідною умовою досяжності із початкової розмітки M_0 розмітки M_k . Коли розмітка M_k досяжна із початкової розмітки M_0 , то система рівнянь $A \cdot x = d$ завжди має розв'язок. Обернене твердження не завжди має місце.

Із необхідної умови досяжності впливає достатня умова недосяжності заданої розмітки в МП. Коли рівняння стану не має розв'язків при заданих початковій і кінцевій розмітках, то кінцева розмітка недосяжна із початкової розмітки в такій МП.

Аналіз структурних властивостей МП. Розглянемо класи МП, зокрема, так звані чисті МП і методи аналізу їх структурних властивостей. Структурні властивості МП – це властивості, які не залежать від початкової розмітки МП в тому сенсі, що вони виконуються для довільної початкової розмітки МП.

Визначення 3. МП (S, W, M_0) називається чистою, якщо з того, що $(p, t) \in F$ витікає, що $(t, p) \notin F$ або $t \cap t^* = \emptyset$.

До структурних властивостей чистих МП відносяться наступні властивості:

- структурна живучість;
- керованість;
- структурна обмеженість;
- консервативність і часткова консервативність;
- повторюваність і часткова повторюваність;

- несуперечливість та часткова несуперечливість.

Визначення 4. МП називається

- структурно живою, якщо існує жива початкова розмітка цієї мережі;
- повністю керованою, якщо довільна розмітка досяжна в цій СП з довільної іншої її розмітки;

- структурно обмеженою, якщо вона обмежена (тобто $\exists k \in \mathbb{N} \forall p \in P: M(p) \leq k$) для будь-якої кінцевої початкової розмітки цієї мережі;

- консервативної, якщо в процесі її функціонування сумарне число фішок в мережі залишається постійним (тобто для будь-яких двох досяжних розміток M і M_1 має місце

$$\sum_{p \in P} M(p) = \sum_{p \in P} M_1(p);$$

- (частково) повторюваною, якщо існує початкова розмітка M_0 МП і послідовність спрацювань переходів σ такі, що кожен (деякий) перехід належить до σ нескінченно часто;

- (частково) несуперечливою, якщо існує початкова розмітка M_0 МП і послідовність спрацювань переходів σ , ведуча з M_0 до M_0 , такі, що кожен (деякий) перехід зустрічається в σ хоча б один раз.

Методи аналізу цих властивостей залежать від класу МП, для яких вони досліджуються. Класифікація МП виконується шляхом введення певних обмежень на МП.

Визначення 5. МП (P, T, F, W, M_0) називається

- машиною станів (МС) $\Leftrightarrow \forall t \in T |t^*| \leq 1$;
- синхрографом (СГ) $\Leftrightarrow \forall p \in P |p^*| \leq 1$;
- вільного вибору (ВВ) $\Leftrightarrow \forall p, p' \in P p \neq p' \rightarrow (p^* \cap p'^* = \emptyset \vee |p^*| \vee |p'^*| \leq 1)$;
- розширеною МП вільного вибору (РВВ) $\Leftrightarrow$

$$\forall p, p' \in P (p \neq p' \rightarrow (p^* \cap p'^* = \emptyset \vee p^* = p'^*));$$

- простою МП (ПМП) $\Leftrightarrow \forall p, p' \in P (p \neq p' \rightarrow (p^* \cap p'^* = \emptyset \vee |p^*| \leq 1 \vee |p'^*| \leq 1))$;

- асиметричною МП (АМП) $\Leftrightarrow \forall p, p' \in P (p \neq p' \rightarrow p^* \cap p'^* = \emptyset \vee p^* \subseteq p'^* \vee p'^* \subseteq p^*)$

Для цих класів мереж, як слідує з визначень, мають місце такі включення:

$$СГ \subset ВВ; МС \subset ВВ; ВВ \subset РВВ \subset АМП; ВВ \subset ПМП \subset АМП.$$

T- і S- інваріанти

Розв'язок x СЛОДР $A \cdot x = 0$ (тобто $M_0 = M_0 + A \cdot x$) називається T-інваріантом МП, а розв'язок y СЛОДР $A^T \cdot y = 0$ називається S-інваріантом МП (або P-інваріантом).

Нехай дана МП з m місцями і n переходами.

Вектор y розмірності m є S-інваріантом МП тоді і тільки тоді, коли $M^T y = M_0^T y$ для довільної фіксованої початкової розмітки M_0 і довільної досяжної розмітки M .

Вектор x розмірності n є T-інваріантом МП тоді і тільки тоді, коли існує розмітка M_0 і послідовність спрацювань переходів σ , яка веде від розмітки M_0 до M_0 такі, що $\sigma = x$.

З теореми про структурну обмеженість чистих МП випливає, що існує вектор y такий, що $A \cdot y \leq 0$. Але тоді існує $x \geq 0$ такий, що $M_0 = M_0 + A \cdot x$ і $M^T y = M_0^T y + x^T A^T y$.

Оскільки $A^T \cdot y \leq 0$, то $M^T y \leq M_0^T y$ і, отже,

$$M(p) \leq \frac{M_0^T y}{y(p)}, \text{ де } y(p) \text{ означає } p\text{-ту}$$

компоненту вектора y .

Ця нерівність дає верхню оцінку для числа фішок, які кладуться в місце p . Ця границя може бути покращена, якщо в цій нерівності використовуються всі S-інваріанти із мінімальної породжуючої множини S-інваріантів. Оскільки інваріанти МП є векторами, то мінімальність векторів розглядається над множиною натуральних чисел N , тобто вектор y називається мінімальним, якщо не існує іншого вектора y_1 такого, що $y_1(p) \leq y(p)$ для всіх місць p .

Породжуюча множина S-інваріантів (T-інваріантів) називається мінімальною породжуючою множиною S-інваріантів (T-інваріантів), якщо не існує жодної не пустої її підмножини, яка теж є породжуючою множиною.

З теорії СЛОДР зрізана множина розв'язків СЛОДР, що відповідає даній МП, є мінімальною породжуючою множиною S-інваріантів (T-інваріантів) і для її побудови можна скористатись TSS-алгоритмом.

Таким чином попередня нерівність зводиться до вигляду $M(p) \leq \min \left[\frac{M_0^T y}{y(p)} \right]$, де

мінімум шукається на всій множині S-інваріантів із мінімальної породжуючої множини.

МП покривається позитивними S- (T-) інваріантами тоді і тільки тоді, коли для довільного місця p_i (переходу t_i) існує S- (T-) інваріант $y \geq 0$ і $y_i > 0$.

Якщо всі місця МП покриваються позитивними S-інваріантами, то МП обмежена.

Наслідок з цієї теореми виглядає таким чином. Якщо існує S-інваріант $y \geq 0$, у якого компонента $y_i > 0$, то місце p_i обмежене, тобто існує натуральне число k таке, що для довільної розмітки $M \in R(S)$ справедлива нерівність $M(p_i) \leq k$.

Якщо МП обмежена і жива, то вона покривається позитивними T-інваріантами.

Якщо МП обмежена і перехід t_i в МП живий, то існує T-інваріант $x \geq 0$, у якого компонента $x_i > 0$.

Пошук дедлоків і пасток в МП.

Розглянемо метод аналізу живучості МП за допомогою дедлоків і пасток, який базується на використанні діофантових обмежень.

Дедлоки і пастки можна знайти в МП за допомогою рішення системи логічних рівнянь, які описують ці властивості, або еквівалентної їй системи лінійних рівнянь над множиною $\{0,1\}$.

Важливість побудови множини дедлоків і пасток для даної МП полягає в тому, що за допомогою аналізу отриманих множин можна визначити, чи жива ця МП чи ні. Це впливає з таких тверджень.

МП вільного вибору як і розширена МП вільного вибору жива тоді і тільки тоді, коли кожен дедлок цієї МП включає пастку, позначену початковою розміткою M_0 .

При використанні МП для верифікації властивостей систем дуже важливим є визначення шляху, що веде до помилки або до підозрілого місця або переходу в МП. Цей шлях прихований в T-інваріантах, проте T-інваріанти містять інформацію лише про те, які переходи спрацьовують, але не несуть ніякої інформації про те, в якій послідовності вони спрацьовують. Якщо встановлено, що МП обмежена, то граф досяжних розміток такої МП є кінцевим автоматом. Застосовуючи до цього автомату алгоритм аналізу, побудуємо регулярний вираз, а по ньому визначимо найкоротші шляхи, що ведуть в підозрілі місця МП.

Технологічний процес дослідження властивостей МП.

Розглянемо такий простий технологічний процес дослідження властивостей МП і представлених ними транзитивні системи.

Спочатку відзначимо, що велика кількість властивостей МП перевіряється засобами лінійної алгебри, зокрема, засобами вирішення систем лінійних рівнянь і нерівностей в області натуральних чисел. Друга група засобів – це засоби побудови та обробки графів і дерев. І третя група – це засоби роботи з мовами (зокрема, з регулярними мовами).

Перша група засобів базується на оригінальному методі побудови мінімальної породжуючої множини рішень систем лінійних рівнянь в області натуральних чисел. Цей метод рішення названий TSS-методом і описаний в [1, 4]. За допомогою цього алгоритму генеруються інваріанти МП без будь-яких додаткових обробок. Використання цього алгоритму дозволяє вирішити проблеми:

- а) досяжності у підкласі МП вільного вибору,
- б) недосяжності розмітки в загальному випадку,
- с) обмеженості МП для єдиного її місця,
- д) обчислення пасток і дедлоків МП,
- е) структурних властивостей для даної МП.

Друга група засобів базується на добре розроблених алгоритмах побудови, представлення та обходу графів [5].

А третя група також має хорошу алгоритмічну підтримку у вигляді алгоритмів аналізу та перетворення регулярних мов і регулярних виразів, результати представлені в [6].

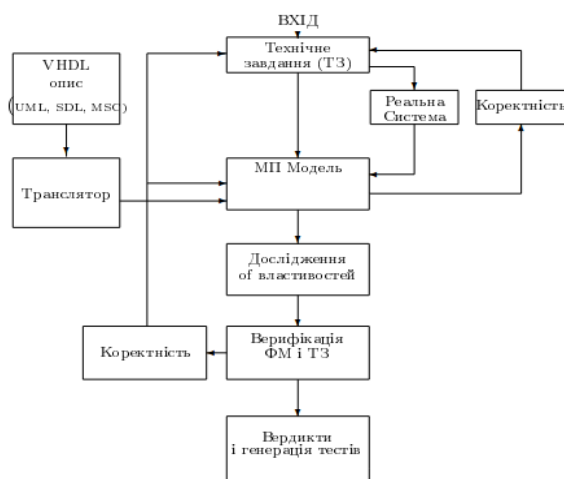


Рис.1. Схема технологічного процесу



Рис. 2. Конкретизація етапів аналізу.

Висновки

Порівняння аналізу МП з логічним підходом до аналізу перерахованих властивостей показує що:

- деякі властивості можуть бути виражені МП більш натурально і більш просто, ніж в логічних формалізмах;
- деякі властивості (обмеженість, можливість повернення до початкового стану, визначення дедлоків і пасток) зовсім не можуть бути виражені або дуже важко виражаються засобами логічних формалізмів;
- перевірка відповідних властивостей засобами МП простіша.

Список використаних джерел

1. Kryvvi S. L. Methods of solution and criteria of consistency of systems of linear diophantine equations over the set of natural numbers. Cybernetics and Systems Analysis. – 1999. – N 4. – P. 12-36.
2. Крывий С. Л., Максимец А. Н. Верификация программ: состояние, проблемы, результаты II. Журнал “Кибернетика і системний аналіз”. 2014, №1, с. 11-20.
3. Hack M. H. T. Decidability Questions for Petri Nets. / Ph. D. Thesis, M.I.T. – 1976.
4. Kryvvi S. A Criteria of compatibility systems of linear diophantine constraints. / Lecture Notes in Computer Science. Spring. Verlag. – 2002. – 2328. – P. 264-271.
5. Jonathan L. Gross, Jay Yellen Handbook of Graph Theory. / CRC Press. – 2004. – P. 57-68.
6. Trakhtenbrot B. A., Ya. M. Barzdin Finite Automata; Behavior and Synthesis. / Nauka. – 1970. – 400 p

Надійшла до редколегії 26.07.13

$$\left\{ \begin{array}{l} a_{11} u_1 + a_{21} u_2 + a_{31} u_3 + \dots + a_{m1} u_m \geq c_1; \\ a_{12} u_1 + a_{22} u_2 + a_{32} u_3 + \dots + a_{m2} u_m \geq c_2; \\ \\ a_{1n} u_1 + a_{2n} u_2 + a_{3n} u_3 + \dots + a_{mn} u_m \geq c_n; \end{array} \right. \quad (5)$$

$$u_i \geq 0, \quad (i = \overline{1, m}) \quad (6)$$

або в матричному вигляді:

$$\min Bu, A^T u \geq C^T, u \geq 0, u = (u_1, u_2, \dots, u_m)^T.$$

Матрична гра у змішаних стратегіях подається як двоїста пара задач лінійного програмування (1)-(6) [1-3], де

$$b_i = 1, \quad i = \overline{1, m}, \quad c_j = 1, \quad j = \overline{1, n}.$$

Задача (4)-(6) є двоїстою до задачі (1)-(3), яку називають прямою.

Пасивність та неактивність стовпців моделі лінійного програмування (1)-(3).

Нехай $X = \left\{ x / \sum_{j=1}^n A_j x_j \leq B, x \geq 0 \right\}$

$$X_r = \left\{ x / \sum_{j=1, j \neq r}^n A_j x_j \leq B, x \geq 0 \right\}$$

Означення 1. Обмеження $a_r x \leq b_r$ пасивне, якщо $X = X_r$.

Означення 2. Стовпець A_r ($x_r = 0$) задачі (1)-(3) пасивний, якщо $X = X_-$.

Аналогічні означення пасивного обмеження та стовпця можна ввести і для задачі (4)-(6).

Теорема 1. [4]. Для того, щоб плани X^* та U^* відповідних спряжених задач були оптимальними, необхідно і достатньо, щоб виконувалися умови доповнюючої нежорсткості:

$$\begin{aligned} x_j^* \left(\sum_{i=1}^m a_{ij} u_i^* - c_j \right) &= 0, \quad j = \overline{1, n} \\ u_i^* \left(\sum_{j=1}^n a_{ij} x_j^* - b_i \right) &= 0, \quad i = \overline{1, m}. \end{aligned}$$

Відомо, що оптимальні стратегії матричної гри визначаються через компоненти розв'язків задач (1)-(3) та (4)-(6) [1]. Складові оптимальних стратегій u_i^* та w_i^* гри пов'язані з компонентами u_i та x_j оптимальних планів двоїстих задач лінійного програмування формулами

$$w_j' = x_j / \sum_{i=1}^n x_i, u_i' = u_i / \sum_{i=1}^m u_i.$$

Наведені вище умови пасивності тісно пов'язані з категорією суттєвості чистої стратегії матричної гри.

Означення 3. Чиста стратегія

$$e_k = \left(\underbrace{0, 0, \dots, 0, 1, 0, \dots, 0}_m \right)$$

називається *істотною* (суттєвою), якщо існує така оптимальна змішана стратегія

$$u' = \left(u_1, u_2, \dots, u_{k-1}, \underbrace{u_k}_{k\text{-а}}, u_{k+1}, \dots, u_m \right), \text{ в якій } k\text{-а}$$

складова u_k додатня. В іншому випадку стратегія e_k *неістотна (несуттєва)*.

Аналогічно визначаються чисті

$$e_k = \left(\overbrace{0, 0, \dots, 0}^k, \underbrace{1, 0, \dots, 0}_v \right) \text{ сущтєві і несущтєві}$$

стратегії другого гравця

$$W' = \left(w_1, w_2, \dots, w_{k-1}, \underbrace{w_k}_k, w_{k+1}, \dots, w_n \right).$$

Введемо згідно [4] основні означення.

Означення 4. Квадратну матрицю A_{σ} , утворену перетином n лінійно незалежних рядків $a_{i_1}, a_{i_2}, \dots, a_{i_n}$ ($i_1, i_2, \dots, i_n$) = $I_{\sigma} \subset I$ та стовпців $A_{j_1}, A_{j_2}, \dots, A_{j_n}$ ($j_1, j_2, \dots, j_n$) = $J_{\sigma} \subset J$ (1)-(3), будемо називати базисною (БМ), а розв'язок $x_0 = (x_{i_1}, x_{i_2}, \dots, x_{i_n})$ відповідної їм системи рівнянь $A_{\sigma} x_{\sigma} = B_{\sigma}$ базисним (БР).

Означення 5. Дві базисні матриці (1)-(3) з одним відмінним рядком будемо називати суміжними.

Базисні матриці в ході ітерацій МБМ [5] послідовно змінюються заміщенням рядків. Нехай e_{r_i} - елементи матриці A_0^{-1} , оберненої до A_0 $\alpha_r = (\alpha_{r_1}, \alpha_{r_2}, ..., \alpha_{r_m})$ - вектор розвинення a_r за рядками базисної матриці A_0 . Введені вектори та їх елементи для МДБМ (коефіцієнти розвинення нормалей обмежень, коефіцієнти оберненої матриці для суміжної “наступної” базисної матриці будемо позначати ризикою зверху, тобто $\bar{A}_0$, $\bar{\alpha}_r$, $\bar{e}_{r_i}$, $\bar{\alpha}_0$.

Теорема 2. [5] Поміж елементами МБМ в суміжних БМ (2)-(3) мають місце співвідношення

$$\overline{\alpha}_{rk} = \frac{\alpha_{rk}}{\alpha_{r_k}}, \quad \overline{\alpha}_{ri} = \alpha_{ri} - \frac{\alpha_{rk}}{\alpha_{r_k}} \alpha_{li},$$

$$r = \overline{0, n+m}; i = \overline{1, n}; i \neq k;$$

$$\bar{e}_{rk} = \frac{e_{rk}}{\alpha_{lk}}, \quad \bar{e}_{ri} = e_{ri} - \frac{e_{rk}}{\alpha_{lk}} \alpha_{li},$$

$$r = \overline{1, n}; i = \overline{1, n}; i \neq k;$$

$$\bar{x}_{0j} = x_{0j} - \frac{e_{jk}}{\alpha_{lk}} \Delta_l, \quad j = \overline{1, n},$$

$$\bar{\Delta}_k = -\frac{\Delta_l}{\alpha_{lk}}, \quad \bar{\Delta}_r = \Delta_r - \frac{\alpha_{rk}}{\alpha_{lk}} \Delta_l,$$

$$r = \overline{1, m+n}; r \neq k;$$

$$C\bar{x}_0 = Cx_0 - \frac{\alpha_{ok}}{\alpha_{lk}} \Delta_l,$$

причому умовою невід'ємності є $\alpha_{lk} \neq 0$.

Відомо, що задача (1)-(3) розв'язна ранг матриці її матриці обмежень п.

Теорема 3. Для того, щоб обмеження (2) $a_r x \leq b_r$ було пасивним, необхідно і достатньо існування базисної матриці A_0 (задачі (1)-(3)), відносно якої компоненти вектору розвинення $\alpha_{rk} \geq 0, k = \overline{1, n}$.

Теорема 4. Для того, чиста стратегія r другого гравця при платіжній матриці A була несуттєвою $x_r = 0$, необхідно і достатньо щоб обмеження (5) $a_r u^T \leq c_r$ було пасивним (щоб існувала базисна матриця A_0 (задачі (4)-(6)), відносно якої компоненти вектору розвинення $\alpha_{rk} \geq 0, k = \overline{1, m}$.

Доведення. Згідно умов теорем 1 про доповнюючу нежорсткість виконуються співвідношення $x_j^* \times \left(\sum_{i=1}^m a_{ij} u_i^* - c_j \right) = 0, j = \overline{1, n}$

$$\text{та } u_i^* \times \left(\sum_{j=1}^n a_{ij} x_j^* - b_i \right) = 0, i = \overline{1, m}. \quad \text{При}$$

виконанні умови пасивності для обмеження (4) двоїстої задачі (4)-(6) буде виконання

$$\sum_{i=1}^m a_{ij} u_i^* - c_j < 0, \quad \forall u^* = (u_1^*, u_2^*, \dots, u_m^*) \in U,$$

витікає, що $x_j^* = 0$.

При виконанні умови пасивності для обмеження (2) прямої задачі (1)-(3)

$$\sum_{j=1}^n a_{ij} x_j^* - b_i < 0, \quad \forall x^* = (x_1^*, x_2^*, \dots, x_n^*) \in X. \text{ Звідки}$$

витікає, що $u_i^* = 0$. Навпаки з рівності $x_j^* = 0$ чи

$$(7) \quad u_i^* = 0 \text{ (при оптимальності та виконанні умов (2)-(3) та (5)-(6)) витікає з теореми, що} \\ \sum_{j=1}^n a_{ij} x_j^* - b_i < 0 \text{ та } \sum_{i=1}^m a_{ij} u_i^* - c_j < 0 \text{ та пасивність.}$$

Для матричної гри у змішаних стратегіях ці положення для елементів методу набудуть наступного вигляду [5].

Твердження 1. Компоненти вектора розкладу цільової функції задачі (1)-(3) в ході ітерацій методу базисних матриць обчислюються

за формулами $\alpha_{0i} = \sum_{j=1}^n e_{ji}, i = \overline{1, n}$, причому для

оптимальності розв'язку необхідно та достатньо, щоб суми елементів стовпців оберненої матриці

$$\text{були невід'ємними } \alpha_{0i} = \sum_{j=1}^n e_{ji} \geq 0, i = \overline{1, n},$$

для єдиності оптимального розв'язку (матричної гри) необхідно та достатньо щоб суми елементів стовпців оберненої матриці були строго

додатніми $\alpha_{0i} = \sum_{j=1}^n e_{ji} > 0, i = \overline{1, n}$, тобто

$$S_1^c > 0, S_2^c > 0, \dots, S_n^c > 0, \quad \text{при}$$

$\exists k_0, \alpha_{k_0} = 0, (S_{k_0}^c = 0)$ розв'язок неєдиний.

Твердження 2. Компоненти вектору розв'язку задачі (1)-(3) в ході ітерацій методу базисних матриць обчислюються за формулами

$$x_{0i} = \sum_{j=1}^n e_{ij}, i = \overline{1, n}.$$

Наслідок 1. Якщо $S_1^r, S_2^r, \dots, S_n^r$ - суми елементів стовпців оберненої матриці $A_0^{-1(r)}$ на r -ій ітерації набувають невід'ємних значень

$$(S_1^r \geq 0, S_2^r \geq 0, \dots, S_n^r \geq 0),$$

то обернена базисна матриця є оптимальною, а відповідний базисний розв'язок є оптимальним, причому його компоненти знаходяться за співвідношенням

$$x_{0i} = \sum_{j=1}^n e_{ij}, i = \overline{1, n} \quad (\text{суми елементів рядків оберненої матриці}).$$

Наслідок 2. Сума елементів рядків оберненої матриці співпадають із значенням відповідної компоненти вектору проміжного розв'язку на ітераціях методу базисних матриць.

Доведення. Справедливість тверджень є наслідком властивостей розкладу елементів методу базисних матриць за рядками базисної матриці.

Неважно переконатись, що при
 $C = (\underbrace{1, 1, \dots, 1}_n); B = (\underbrace{1, 1, \dots, 1}_m)^T$ та відомими

$A_b, A_b^{-1} \quad \alpha_0 = C \times A_b^{-1}, x_0 = A_b^{-1} \times B_b$ для формул
зв'язку елементів методу базисних матриць (7)-
(9) будуть справедливі наступні співвідношення

$$\alpha_{0i} = \sum_{j=1}^n e_{ji}, i = \overline{1, n}, \quad x_{0i} = \sum_{j=1}^m e_{ij}, i = \overline{1, n}.$$

При розв'язанні двоїстої пари задач
лінійного програмування (матричної гри у
змішаних стратегіях) на основі методу базисних
матриць можна будувати нові алгоритми з
врахуванням встановлених властивостей.

Схема методу допустимих базисних матриць

Розвинення положення теореми 1. для
аналізу лінійної моделі (1)-(3) при існуванні
допустимої початкової базисної матриці:

Між коефіцієнтами розвинення нормалей
обмежень (2)-(3) та цільової функції (1),
елементами обернених матриць, базисними
розв'язками, нев'язками обмежень (2)-(3) та
значеннями цільової функції в двох суміжних
допустимих базисних розв'язках мають місце
співвідношення (7)-(11) причому умовою:
опорності базисної матриці при вводити вектору
нормалі a_i обмеження $a_i x \leq b_i$ на k -ту позицію
базисної матриці A_b є виконання $\alpha_{ik} \neq 0$,
умовою допустимості опорного базисного
розв'язку є $\alpha_{ik} < 0$, зростання цільової функції
 $\alpha_{0k} < 0$, спадання - $\alpha_{0k} \geq 0$, сталості $\alpha_{0k} = 0$.

Приведені твердження охоплюють всі
випадки для організації розширеного аналізу
лінійної системи (моделі матричної гри) на
основі методу допустимих базисних матриць:

1. Якщо існує базисна матриця A_b така, що
 $\alpha_{0k} \geq 0, k = \overline{1, n}$, то базисна матриця та
відповідний їй розв'язок x_0 оптимальні, причому
при $\alpha_{0k} > 0, k = \overline{1, n}$, розв'язок єдиний, при
 $\exists i_0, \alpha_{0i_0} = 0, i_0 \in I$, розв'язок неєдиний.

2. Якщо існує k для якого $\alpha_{0k} < 0$ та $\alpha_{rk} \geq 0$
для всіх небазисних r , то цільова функція задачі
необмежена на множині допустимих розв'язків.

4. Якщо існує k для якого $\alpha_{0k} < 0$ та $\alpha_{rk} < 0$,
то з допомогою перетворень (7)–(11) можна
перейти до нової базисної матриці та опорного
розв'язку з більшим значенням цільової функції.

5. Якщо існує базисна матриця A_b та
розв'язок u_0 такі, що $\alpha_{rk} \geq 0, k = \overline{1, m}$, то
обмеження $a_i u^T \leq c_i$ пасивне.

Висновок

Умови пасивності обмежень (2)-(3)
встановлені в середовищі МБМ дозволяють
застосуванням теореми про доповнюючу
нежорсткість визначати пасивні стовпці задачі
(4)-(5).

Встановлений зв'язок пасивності обмежень
двоїстої та прямої задач з властивістю
несуттєвості чистих стратегій гравців дозволяє
аналізувати структуру платіжної матриці гри.

Список використаних джерел

1. Golshteyn E.G., Yudin D.B. New directions in linear programming. – M. – Sovetskoe radio, – 1969, – 524p. (in Russian).
2. Dantzig G.B. Linear programming and application. M.: Progress, – 1966. (in Russian).
3. Dantzig G.B., Thapa M.N. Linear Programming 1: introduction, Springer, – 1997, – 435p.
4. Golshteyn E.G., Yudin D.B. Linear programming/. Theory and methods. – M.: Nauka, – 1963. – 776p. (in Russian).
5. Kudin V. I., Lyashko S.I., Khritonenko N.V., Yatsenko Yu.P. Analysis of the properties of a linear system using the method of artificial basis matrices // Kibernetika i sistemny analiz. — 2007. — N 4. — P. 119–127 (in Ukrainian).

Надійшла до редколегії 19.06.13

УДК 681.3.06

Лавріщева К.М., д.-р ф.-м. н., проф. *

Підхід до формального подання онтології життєвого циклу програмних систем

Запропоновано застосування онтологічного підходу до формального опису життєвого циклу (ЖЦ) програмних систем (ПС) з метою реалізації семантики його процесів для подальшої генерації варіантів ЖЦ для виготовлення різних видів ПС. Обґрунтовано засоби онтології щодо подання моделі життєвого циклу стандарту ISO/IEC 12207, включаючи основні, організаційні та керуючі класи процесів. Отримано графічне представлення моделі ЖЦ засобами предметно-орієнтованої мови DSL та правилами інтелектуалізації знань в системі Protégé. Для процесу тестування надано онтологічний опис понять в термінах Protégé і його семантики в мові програмування Ruby. Кінцевий результат опису онтології ЖЦ подано в мові XML. Отриманий опис ЖЦ і програму тестування відображено у інструментально-технологічному комплексі (ITK) веб-сайту <http://sestudy.edu-ua.net> технологічною лінією.

Ключові слова: онтологія, модель життєвого циклу, стандарт ЖЦ, процеси, тестування, DSL, опис, система Protégé, комплекс ІТК.

*Інститут програмних систем НАН України,
03680, м. Київ-187, просп. Ак. Глушкова 40
E-mail: lavryscheva@gmail.com

K.M. Lavrischeva, Dr. Sci. (Phys.–Math.), prof.*

Approach to the formal presentation of ontology of life cycle of the program systems

Application of ontological approach is offered to the formal specification of Life Cycle (LC) of the program systems (PS) by the ontology facilities with purpose of realization of semantics of processes for the subsequent generation of the LC variants for making the different kinds PS. Ontological approach in relation to presentation of model of LC of the ISO/IEC 12207 standard, including basic, organizational and handling classis of processes was done. Presentation LC of the grafic facilities of the subject-oriented DSL and roles of mining knowledge areas in the Protégé system are shown. For testing process is given ontological description of notions tasks in terms of Protégé systems and its semantics in language programming Ruby. Accordingly this description of ontology LC is conducted in language XML. The *got* description LC and program of testing is done in the instrumental-technological complex (ITC) web-site <http://sestudy.edu-ua.net> of technological line.

Key Words: ontology, model of life cycle, the LC standard, processes, testing, DSL description, the Protégé system, the ITC complex

*Institute of Software Systems NAS Ukraine,
03680, Kyiv-187, 40, Glushkov Ave.
E-mail: lavryscheva@gmail.com

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

1. Вступ

В даній роботі пропонується нова концепція автоматизації загальних процесів життєвого циклу (ЖЦ) і генерації варіантів спеціалізованих ЖЦ для їх використання в сучасних прикладних, програмних і інформаційних технологіях оброблення даних та виконання в розподілених новітніх середовищах Grid, Cloud, високопродуктивних кластерних систем та в семантичних вебах Інтернету. Ця концепція сформульована автором в рамках теми фундаментального проекту ІПС НАНУ (2011–

2016) щодо створення теоретичних і прикладних аспектів технології розроблення розподілених прикладних систем з готових ресурсів і веб-семантик. Дана тема є подальшим розвитком технології зборочного програмування інституту у напрямку автоматизованого промислового збирання програмних продуктів (ПП) як на збиральних конвеєрах фабрик програм ідеї Глушкова з використанням накопичених готових компонентів повторного використання (КПВ), а саме, з модулів, об'єктів, компонентів, програм, сервісів, процесів, рішень – аскетів тощо [1, 2].

Стандартний ЖЦ ISO/IEC 12207–2007 є могутнім загальним механізмом побудови різноманітних ПП. До його складу входять 17 процесів, 74 під процесів (дій) та 232 технологічних задачі. Автоматизація ЖЦ це дуже складна і важка проблема. Вона зараз вирішується багатьма фірмами в загальних системах фрагментарно і методично. Автомат відсутній. Запропонована нами концепція автоматизації ЖЦ через формальну модель ЖЦ є спробою розвитку майбутній індустрії ПП [3].

Крім того, для виконання цієї концепції створені нові мови опису концептуальних моделей знань: OWL (Web Ontology Language), XML (Extensible Markup Language), ODS (Ontology-Driven Software Development) BPL, MBPN (Modeling Business Process Notation) тощо. Є системи моделювання предметних областей (ПрО), доменів – ODM (Organizational Domain Modeling), FODA (Feature-Oriented Domain Analysis), DSSA (Domain-Specific Software Architectures), Eclipse-DSL, DSL (Domain Specific Language) Tools VS.Net, Protégé тощо. Тобто склалися умови для формального опису процесів ЖЦ і моделювання на них різних ПП.

Використовуючи відповідні онтологічні засоби для опису процесів ЖЦ новою мовою MBPN, предметно-орієнтованою мовою DSL (Domain Specific Language), а також мов опису семантики цих процесів, автор пропонує підхід к реалізації сформульованої концепції. В якості приклада реалізації даної концепції обрано процес тестування шляхом моделювання його під процесів і дій онтологічними засобами і опису семантики мовцю програмування. При розробки домену програмної інженерії ЖЦ ПС використані онтологічні інструментальні засоби, DSL Tools VS.Net і Protégé [4, 5].

Важливим аспектом моделювання різних доменів є понятійна база і систем понять, за допомогою якої формуються усі проблеми доменів. Понятійна база відображає термінологію, суттєві відношення між поняттями та їхню інтерпретацію. Серед відношень головними є такі:

- узагальнення для звуження істотних ознак понять, розширення їх коло та обсягу;
- конкретизація з додаванням істотних ознак для розширення понять і звуження їх обсягу;
- агрегація, як об'єднання понять у нове поняття, істотні ознаки якого можуть бути сумою ознак понять або суттєво новими;

– асоціація, що стверджує наявність зв'язку між поняттями без уточнення залежності їх від змісту та обсягів.

Сукупність термінології, понять, характерних для них відношень та парадигми їхньої інтерпретації у межах домену прийнято називати *онтологією доменних знань*.

У самому загальному випадку, *онтологія* – це угода про спільне використання понять, що включає засоби подання предметних знань і домовленості про методи міркувань.

Онтології представляються семантичними мережами, вузлами яких є поняття, а дугами – зв'язки чи відношення, асоціації між ними. На даний час онтологічні підходи отримали широке розповсюдження у вирішенні проблем подання знань, семантичної інтеграції інформаційних ресурсів, інформаційного пошуку тощо. Вони дозволяють одержувати описи класів об'єктів ПрО, що відображають поняття та знання про них. Онтології деяких ПрО подаються знаннями, словниками понять, концептів та відношень між ними. Доречі, мова XML стала стандартом розмітки різноманітних даних ПрО для їх зберігання і обміну між різними доменами. Вона є засобом автоматичної трансформації описів моделей ПрО в сучасних онтологічних мовах до схем, придатних для роботи в різних прикладних застосуваннях.

Запропонована концепція розглядалася в Київському національному університеті імені Тараса Шевченка на наукових семінарах кафедр ТТП і ІС, а також при викладанні дисципліни «Програмна інженерія». В рамках цієї дисципліни студенти вивчають сучасні методи і засоби моделювання ПрО і побудови ПС, а також вивчаються сучасні стандарти ЖЦ ISO/IEC 12207 –2007 та загальних типів даних GDT (General Data Types) ISO/IEC 11404 – 2006. На практичних заняттях проводиться апробацію засобів загального опису і реалізації деяких експериментальних онтологій: DSL Tools VS.Net та Protégé. На прикладі опису окремих фрагментів деяких доменів онтологічними засобами побудовані моделі ПрО з метою їх використання при побудові ПП [2]. Деякими студентами кафедри ІС і ТТП факультету кібернетики захищені дипломні роботи по даній тематиці з використанням онтологічних засобів та понять (класів, аксіом, слотів, фасетів тощо).

В роботі пропонується підхід к автоматизації ЖЦ і його апробація студентами в плані вивчення стандарту ЖЦ, його структури, процесів і дій, а також використання засобів онтології для їх опису і реалізації на відкритих онтологічних інструментах – DSL Tools VS.Net і Protégé [4 -6].

2. Онтологія загального життєвого циклу програмних систем

ЖЦ отримав еволюційний розвиток, починаючи від спрощених ЖЦ на кожній прикладній системі, моделей водоспадної, спіральної ітераційної тощо. Вони сформували окремі і загальні процеси при розробці різних типів і видів програмних продуктів (ПП). В результаті був створений стандарт ISO/IEC 1996 (перша редакція), потім в 2007 його друга редакція ЖЦ, яка відображає загальну структуру процесів, що можуть приймати участь при розробленні ПП.

2.1 Подання формального опису ЖЦ стандарту ISO/IEC 12207–2007

Процеси ЖЦ подані у стандарті у трьох категорій [5]:

- основні процеси;
- процеси підтримки;
- організаційні процеси.

Для кожного з процесів визначені види діяльності (дії – activity), задачі, сукупність результатів (виходів) діяльності і розв'язання задач, а також деякі специфічні вимоги. У стандарті наведено перелік робіт для основних, організаційних процесів і процесів підтримки, але не спосіб їх виконання і не форма подання результатів.

До основних процесів належать:

- процес придбання, визначає дії покупця при отриманні автоматизованої системи або сервісу. Дії такі: ініціювання і підготовка запиту, оформлення контракту; моніторинг і приймання;
- процес постачання, визначає дії з передачі продукту або сервісу покупцю. Це такі: підготовка пропозицій, оформлення контракту, планування, виконання і контроль продукту, а також його оцінка і постачання;
- процес розроблення визначає процеси і дії для розробки продукту (рис.1);
- процес експлуатації, визначає дії оператора з обслуговування системи в ході її експлуатації користувачами (рис 1);

– процес супроводження, який визначає дії організації, що виконує супровід ПП (керування модифікаціями, підтримку поточного стану і функціональної придатності, інсталяцію ПП в операційному середовищі, супроводження і модифікацію, розробку планів модифікації системи, міграції ПП в інше середовище тощо. Стандарт ЖЦ містить у собі опис допоміжних процесів, що регламентують додаткові дії з

перевірки продукту, керування проектом та його якістю (рис.2).



Рис. 1. Схема основних процесів ЖЦ ПС

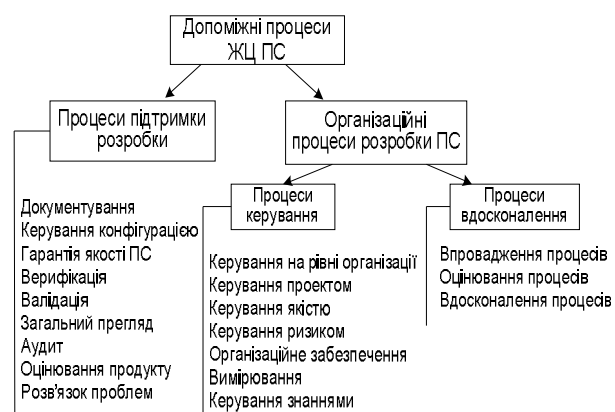


Рис. 2. Схема допоміжних процесів ЖЦ ПС

Відповідно рис. 2, процес підтримки розробки містить: документування, керування версіями, верифікація і валідація, перегляди, аудити, оцінювання продукту та ін. До організаційних процесів належать: керування проектом (менеджмент розробки) та вдосконалення процесів. Процес керування включає процеси (рис.2) керування конфігурацією, проектом, якістю, ризиками, змінами тощо. Процес вдосконалення включає впровадження, оцінку проекту та його вдосконалення.

Кількість процесів, дій і задач стандарту ЖЦ наведені в Табл. 1.

Таблиця 1. Процеси, під процеси і дії ЖЦ.

Класи	Процес	Дія	Задача
Основні процеси	5	35	135
Процеси підтримки	8	25	70
Організаційні процеси	4	14	27
Всього	17	74	232

Залежно від цілей конкретного проекту головний розробник і менеджер вибирають процеси, дії і задачі, вибудовують визначену схему ЖЦ для застосування в конкретному програмному проекті.

Опис семантики процесів, парадигм і методів їх виконання (об'єктні, компонентні, сервісні й ін.) наведені у ядрі знань SWEBOK (www.swebok.com) та у підручнику [2].

В кожній технології програмування ПС з використанням стандарту ЖЦ застосовуються теоретичні, прикладні методи, стандарти якості, загальні і фундаментальні типи даних (ISO/IEC 15404, ISO/IEC 9126, ISO/IEC 11404 GDT та ін.), а також рекомендації і методики цих стандартів. Задача автоматизації загального, стандартного ЖЦ виникло у відділі ІПС НАНУ у зв'язку зі створенням системи автоматизованого подання ЖЦ для різноманітних розподілених ПС (РПС) і генерацію різних варіантів ЖЦ при виготовленні окремих компонентів РПС, включаючи ПС із готових ресурсів, нових програмних компонентів та сервісів.

Враховуючи цю задачу, автор при викладанні нормативного курсу студентам КНУ, проводив додаткове вивчення студентами особливостей стандартів та механізмів їх подання в сучасних операційних середовищах. На практичних заняттях студенти вивчали засоби візуального подання процесів ЖЦ за допомогою WWF (Windows Workflow Foundation) та DSL Tools VS.Net. На основі цих засобів студенти виконали опис онтології ЖЦ в графічному і XML видах в рамках систем DSL Tools VS.Net і Protégé.

2.2 Подання онтології моделі ПрО ЖЦ і моделі характеристик

Дамо опис моделі для домену ЖЦ стандарту описаного вище з процесів Р (process), дій А (Action) і задач Т (Task).

Моделі ЖЦ має такий вигляд:

$$M_{жц} = (P_{\kappa}, A_m, T_n),$$

де $P_{\kappa} = (P_{1\kappa}, P_{2\kappa1}, P_{3\kappa2}) - P_{1\kappa}, \kappa = 1-5$ (основні процеси ЖЦ першої категорії), $P_{2\kappa1}, \kappa1 = 1-8$ (процеси ЖЦ підтримки другої категорії), $P_{3\kappa2}, \kappa2 = 4$ (організаційні процеси ЖЦ третьої категорії);

$A_m = (A_{\kappa r}, A_{\kappa1l}, \dots, A_{\kappa2j})$, де $A_{\kappa r}, r = 1-35$ (дії на основних процесах ЖЦ), $A_{\kappa1l}, l = 1-25$ (дії на процесах підтримки ЖЦ), $A_{\kappa2j}, j = 1-14$ (дії на організаційних процесах ЖЦ);

$T_n = (T_{n\kappa}, T_{nl}, T_{nj}) - T_{n\kappa}, \kappa = 1-135$ (задачі основних процесів ЖЦ), $T_{nl}, l = 1-70$ (задачі процесів підтримки ЖЦ), $T_{nj}, j = 1-27$ (задачі організаційних процесів ЖЦ).

Зміст процесів, операцій поданий вище міститься в сутності, а опис змісту задач по ним наведено в самому стандарті. Задачі не формалізовані і надалі спочатку буде дано опис

їх призначення, потім підбір мов для формального опису для реалізації їхньої семантики.

Для подання структури ЖЦ використана графічна мова DSL. Ця мова має виразну особливість, спрямовану на відображення процесної специфіки ЖЦ, тоді як мови загального призначення (Java, C++, C# і ін.) орієнтовані на опис дій будь-яких програм обробки даних. Мова DSL містить загальні абстракції для відображення класів об'єктів ПрО типу процес, дія, а також відношень між ними [3]. По своєму змісту ця мова близька до мов HTML, XML, WSDL тощо.

Моделі ПрО, описано за допомогою однієї мови DSL, можуть бути трансформовані у моделі за допомогою іншої DSL. Це дозволяє вільно інтегрувати між собою різні частини процесів системи, написані на різних мовах DSL. Тобто ПрО ЖЦ може бути описана на одному рівні абстракції, а потім перетворена з додатковою деталізацією на більш низький рівень абстракції, що дозволяє доповнювати модель ПрО повторними компонентами і об'єктами. Головною компонентою моделі ПрО ЖЦ є модель загальних характеристик процесів як об'єктів ПрО.

Моделі ЖЦ ПрО може бути автоматизована з використанням високоточних доменно-специфічних мов, налаштованих спеціально на процеси і дії, що є в класі мов онтологій. Моделі можуть містити інформацію про об'єднання процесів і дій, включаючи артефакти, що беруть участь в ній, а також їх залежність між собою. Вони також можуть містити інформацію про конфігураційну структуру програм оброблення процесів, апаратних і програмних ресурсів, необхідних при виконанні програм автоматизації процесів та їх розгортання.

Опис онтології моделі характеристик ПрО

Передумову розробки DSL становить детальний аналіз та структурування домену ПрО. Серед існуючих методологій доменного аналізу найбільш відомі такі: ODM (*Organization Domain Modeling*), FODA (*Feature-Oriented Domain Analysis*), характеристичний аналіз домену та DSSA (*Domain-Specific Software Architectures*).

При аналізі домену створюється модель характеристик, яка забезпечує узагальнення і розбіжності процесів ПС в домені шляхом індикації загальних характеристик для усіх процесів ПС та відмінних характеристик кожного з процесів ЖЦ.

Моделі характеристик подається діаграмами характеристик з описом залежностей між ними. Концепція діаграм характеристик успадкована з методу FODA. Вона дає можливість стисло описувати

усі можливі конфігурації процесів в межах різних категорій процесів ЖЦ, що розглядаються як *екземпляри* або *зразки* (instances), виділяючи загальні та альтернативні характеристики, які можуть бути відмінними для кожній з конфігурації процесів ЖЦ.

На даний час нотація діаграми характеристик виконується мовою DSL під назвою FDL (Feature Definition Language), як мови опису характеристик поняття домену і формально визначених операцій для обробки виразів FDL;

Діаграми характеристик подають системні характеристики. При створенні автоматизованих інструментів, призначених для побудови діаграм характеристик і їх обробки, потрібно текстове подання. Воно завдає усю інформацію, що існує у графічній діаграмі. Визначення складається з множини *характеристик* (feature definitions), імен характеристик та *виразу* (feature expression), що включає:

- атомарні характеристики;
- композиційні характеристики: імена яких визначенні у іншому місці;
- необов'язкові характеристики (*optional*) виразу, що завершується символом “?”;
- обов'язкові характеристики (*mandatory*) виразів, які замкнені у конструкції *all()*;
- альтернативні характеристики (*exclusive* – вибір) у виразу *one-of()*;
- невиключного набору характеристик (“or-features”) з переліку характеристикних виразів *more-of()* й їх комбінації;
- значення характеристики за умовчанням (default) – атомарної характеристики;
- інші (невизначені) характеристики у формі “пусто...”.

Специфікація характеристик SDF надає формалізми для визначення синтаксису, який можна порівняти з формою BNF (П.Наура) для проведення лексичного і семантичного аналізу описаних характеристик моделі ПрО, яка використовується при створенні різних варіантів ПС.

Механізми залежності характеристик

Запропонований підхід міститься на принципі успадкування характеристик з такими умовами:

- кожна характеристика відповідає класові;
- *асоціації* (зв'язки) між класами помічені так званим *<стереотипом>*, який позначає вид залежності характеристики;
- обов'язкова (*mandatory*) залежність між *агрегацією* в класах;

– необов'язкова (*optional*) залежність між *асоціацією* і діапазоном кардинальних чисел (потужністю множини чи кількістю елементів множини) від 0 до 1;

– обов'язковий перелік *one-of* та *more-of* в специфікованих підкласах кожної з альтернатив.

Результат трансляції опису характеристик в FDL може бути поданий мовою XML, як формату обміну інформацією Meta data (the XML Meta data Information Exchange format). XML-документи можуть бути імпортовані в інструменти моделювання UML, такі як Rational, TogetherJ та ArgoUML, а також для генерації класів Java. Після створення мови DSL для домену необхідно використати мову FDL. Підхід до опису моделі характеристик ПрО використано при розробленні варіантів ЖЦ ПС шляхом конфігурування різних процесів системи автоматизації ЖЦ. На дану модель поважається задача забезпечення генерації варіантів спеціальних ЖЦ з включенням необхідних процесів для реалізації заданого класу ПС і заміни чи додавання семантики деяких задач для нових процесів. Для отримання робочого варіанта ЖЦ ПС планується використання засобів Java [6, 7].

2.3 Онтологія стандарту ЖЦ в DSL Eclipse

Іншим підходом для опису онтології доменів є Eclipse DSL. Це середовище для розробки графічних моделей ЖЦ. Дане середовище є одне з самих ефективних інструментів для розробки доменів на мові DSL.

На самому початку необхідно розробити візуальну модель проекту. Середовище надає інструменти для опису класів розділів домену ЖЦ та відношень між ними (рис. 3, 4).

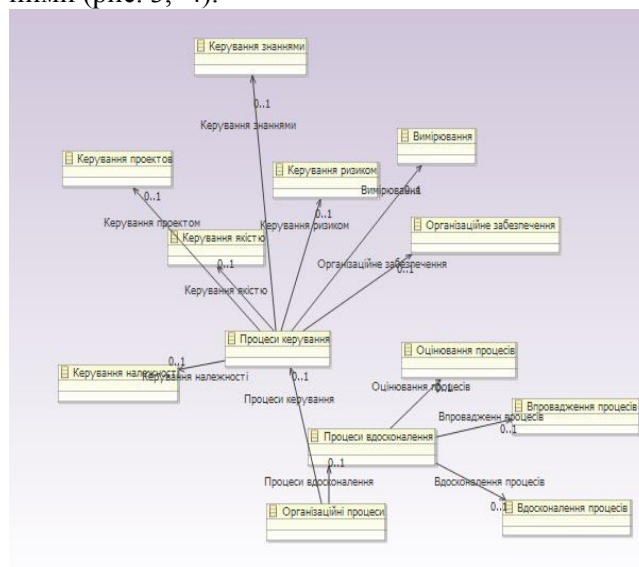


Рис.3. Онтологія основних процесів ЖЦ

Типи відношень дозволяють реалізувати основну логіку проекту. В кожному класі описуються наявні методи та поля необхідні для функціонування проекту. Процеси підтримки містять в собі всі процеси, що виконуються після побудови проекту та підтримують його працездатність та актуальність. Їх онтологічна структура відповідає структурі основних процесів і наводиться не буде.

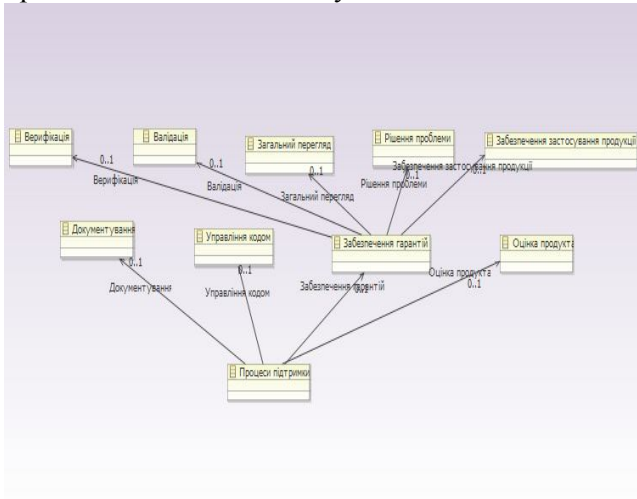


Рис. 4. Структура онтології організаційних процесів ЖЦ

Наступним кроком є генерація текстового представлення наявних графічних моделей, а потім генерація в XML-мову. Процес тестування ЖЦ анотований засобами подання знань про поняття і їх відношення у Protégé.

Текстовий опис процесів ЖЦ мовою XML

Дане графічне подання процесів ЖЦ було використане для отримання тексту в мові XML.

Помилки в графічному описі, що були знайдені дизайнером і виправленні за допомогою відповідного редактору. Після цього результат кожного процесу подається в XML. Нижче наведено приклад фрагменту опису фрагменту головних процесів ЖЦ в мові XML.

```
<?xml version="1.0" encoding="utf-8"?>
<AssociationLine Name="Визначення_вимог"
Type="Main.Визначення_вимог" ManuallyRouted =
"true" FixedFromPoint = "true" </AssociationLine>
<AssociationLine Name = "Інтеграція_ПС" Type =
"Main.Інтеграція_ПС"
<AssociationLine Name = "Інсталяція" Type =
"Main.Інсталяція"
<AssociationLine Name = "Експлуатація" Type =
"Main.Експлуатація"
<Property Name = "Інтеграція_ПС" />
<Property Name = "Інсталяція" />
<Property Name = "Аналіз_вимог" />
<Property Name = "Експлуатація" />...
```

Після отримання описів процесів в XML їх семантичний опис. На прикладі процесу тестування ЖЦ виконано анотування засобами Protégé. Це новий вид опису, який відсутній в практиці програмування і орієнтований на тестування сучасних програм [1–3].

3 Опис онтології процесу тестування ЖЦ

Концептуальна модель процесу тестування ПС має вигляд [4]: $SFT = \langle TM, TD, TA, Env \rangle$,

де TM – під процес керування тестуванням;

TD і TA – під процеси тестування відповідно Про та застосувань;

Env – концептуальне й інформаційне середовище процесу тестування ПС.

Усім трьом під процесам дамо уніфіковане формальне подання:

$TM = \langle Task(TM), En(TM), CM(TM) \rangle$,

$TD = \langle Task(TD), En(TD), CM(TD) \rangle$,

$TA = \langle Task(TA), En(TA), CM(TA) \rangle$,

$En(TM) \cup En(TD) \cup En(TA) = Env$,

де $Task$ – задачі відповідного під процесу;

En – концептуальне й інформаційне середовище відповідного під процесу;

CM – під модель координації операцій відповідного під процесу.

Склад середовища Env визначається виразом

$Env = TG \cup SG \cup T \cup P \cup RG \cup RP$,

де TG і SG – тестові активи і готові програми;

T і P – тести і тестовувані за стосунки;

RG і RP – звіти про виконання тестових програм і тестів.

Онтологічний опис процесу. Для опису процесу тестування використана онтологічна система Protege. В ній знання про модель процесу подаються класами, слотами, фасетами та аксіомами. Подібну можливість надають також і інші інструменти. Наприклад, діаграми класів в UML системи Rational Rose, які можуть відображатися в програмний код на декількох мовах програмування.

Для подання онтології тестування виділяються дві групи понять: *прості* і *складні*.

Прості поняття тестування. Це такі: Тестер (*Tester*), Контекст (*Context*), Дія (*Activity*), Метод (*Method*), Артефакт (*Artefact*) і Середовище (*Environment*).

Просте поняття можуть мати атрибути. В якості *атрибутів* вибрані такі під поняття, які характеризують базове (батьківське) поняття і можуть приймати конкретні значення.

Дамо короткий зміст основних понять.

Tester – визначає суб'єкт або об'єкт, який виконує тестування. Група тестування має лідера, який є

атрибутом поняття, а його ім'я – значенням атрибута. Атрибутами є ім'я, тип, обов'язки. Атрибут тестера – обов'язки – описує, що може робити тестер в процесі тестування. Поняття обов'язки – складне поняття, яке визначається на основі простих понять. Для цього поняття можна виділити наступні атрибути: ім'я (tester_name), тип (tester_type), обов'язки. Атрибут тестера – обов'язки – описує дії, які може робити тестер в процесі тестування.

Приклад фрагменту XML-документа тестера:

```
<!-- TESTER -->
<xs:element name = "TESTER">
  maxOccurs = "unbounded" ref = "TESTER"/>
  <xs:sequence>
    <xs:attributename = "TESTER_TYPE" use =
"required">
      <xs:simpleType>
        </xs:attribute>
      <xs:attributename = "TESTER_NAME" use =
"required"/>
```

Контекст визначає відповідні рівні, методи тестування, входи і виходи задач тестування. В онтології це поняття визначає один атрибут: Context_type (Рівень_тестування) за формою рівень_тестування = {модульне, інтеграційне, системне, регресійне}

Дія складається з понять, що деталізують кроки процесу тестування: планування тестування, розроблення (генерація) тестів, виконання тестів, оцінка результатів, вимірювання тестового покриття, генерація звітів тощо. Для цього поняття завдається один атрибут – тип дії (Activity_type) з можливими значеннями: тип_дії = {планування, розробка тестів, виконання тестів, перевірка результатів, оцінка покриття, підготовка звіту}

Метод – це поняття, якому відповідає декілька методів тестування. Наприклад, для модульного тестування – методи структурного і функціонального тестування. Кожний метод по відношенню до початкового коду може класифікуватися як «біла скринька», «чорна скринька» або базоване на специфікації тестування (specification-based). Фрагмент XML-схеми поняття методу:

```
<!-- METHOD -->
<xs:element name = "METHOD">
  <xs:complexType>
    <xs:attribute name = "METHOD_NAME" use =
"required">
      <xs:simpleType>
        <xs:restriction base = "xs:token">
          <xs:enumeration value =
"CONTROL_FLOW_TESTING"/>
```

```
<xs:enumeration value =
"DATA_FLOW_TESTING"/>
<xs:enumeration ...
</xs:enumeration>
```

Методи, базовані на коді, підрозділяються на: структурні; підсіву помилок; мутаційні. Структурні методи підрозділяються на: тестування потоку керування і тестування потоку даних. Методи тестування потоку керування (control-flow methods) включають покриття операторів, покриття гілок і різні критерії покриття шляхів. Ці конкретні методи тестування є екземплярами різних підкласів методів тестування.

Аналогічним чином класифікуються методи «чорної скриньки» або засновані на специфікації: функціональні; на припущенні про помилки; евристичні і ін.

З іншого боку, по відношенню до процесу пошуку помилок і відмов, всі методи можна розділити на систематичні (пошуку помилок) і стохастичні (статистичні) – виявлення відмов.

Таким чином, для опису методу тестування введемо наступні атрибути:

- ім'я (назва методу), «розбиття на категорії»;
- тип методу (структурний, базований на помилках);
- підхід, базований на коді, на специфікаціях, статистичний.

Таке розбиття методів дозволяє однозначно класифікувати кожний метод тестування і розширювати онтологію.

Артефакт. Кожна дія з тестування може включати декілька артефактів, таких як об'єкт тестування, проміжні дані, результати тестування, плани, набори тестів, скрипти тощо. Їх називають «тестовими активами». Об'єкти тестування можуть бути різних типів: початковий код, HTML файли, XML файли, вбудовані зображення, звук, відео, документи і ін. Всі ці артефакти відображаються у онтології. Кожний артефакт також асоціюється з місцем свого зберігання, даними, історією створення і перегляду, наприклад, творця, час оновлення, номер версії тощо.

Enviroment. Програмне середовища, де виконується тестування, як правило матується такими даними: ім'я, тип і версія продукту. Дане поняття розбивається на два під поняття: апаратне і програмне з атрибутами. Атрибутами апаратного середовища є: назва пристрою, модель, виробник.

Атрибутами програмного середовища є: назва продукту, тип продукту і версія. Можливими значеннями атрибута можуть бути такі:

Середовище = {ОС, БД, Компілятор, Веб-браузер}

Складні поняття процесу тестування. До складних понять відносяться такі: обов'язки

доповіді на конференції ICTERI–2012 і 2013, в яких відображені нові підходи до навчання студентів, що вивчають університетах Computer Sciences.

5. Висновки

Запропоновано нову концепцію автоматизації ЖЦ стандарту ISO/IEC 12207–2007 і підхід до її реалізації з використанням сучасних онтологічних засобів (Eclipse DSL та Protégé) подання процесів і систем моделювання ПрО. Отриманий нами досвід використання онтологічного підходу, який широко поширений при поданні онтологій у середовищі Internet, наприклад знання про різні наукові дисципліни e-sciences (медицина, біологія, хімія, генетика, Computer Science тощо) показує можливість онтологічного подання важливої предметної області наукової дисципліни «Software Engineering» з прикладом реалізації процесу тестування ЖЦ засобами мови програмування Ruby, а також інших областей знань, як обчислювальна геометрія, загальні та фундаментальні типи тощо. Вони експериментально реалізовані і подані в ІТК для їх застосування в процесі електронного навчання студентів новим дисциплінам комп'ютерних наук і методам реалізації задач автоматизації прикладних систем з використанням технологічних засобів.

Список використаних джерел

1. Lavrischeva E., Hrischenko V. The assembly programming. Basics of the software industry.– Second Ed.– K. Naukova Dumka, 2009.–371p. (In Russian)
2. Lavrischeva E. Implementing the ideas of Academician VM Glushkov on Software Engineering, Visnik NAS Ukraine, 2012, №3. – P. 67–79. – Available at http://www.nbu.gov.ua/portal/all/herald/2013_3/ (In Ukrainian)
3. Lavrischeva E. Ontological representation for the total life cycle of software line of software products. - Proceedings TAAPSD'2013, «Theoretical and practical aspects of building software systems», 25.05 – 2.06.2013.– P.81–90. (In Ukrainian)
4. Lavrischeva E., Koval G., Babenko L., Slabospitska A., Ihnatenko P. New theoretical foundations technology families of software systems in the context of generating programming – Site, 2012. – Available at <http://www.nbu.gov.ua/> (In Ukrainian)
5. Lavrischeva E., Software Engineering. – Textbook. – K.: Akademperiodika, 2008. – 319 p. (In Ukrainian)
6. Gromback R. Eclipse Modelling Project. A Domain Specific Language (DSL) Toolkit, Addison Wesley, 2009.–232 p.
7. Lavrischeva E. Instrumental Institution to develop and study techniques of software systems Вісник NAS Ukraine, 2012, №3. – P. 67–79. – Available at http://www.nbu.gov.ua/portal/all/herald/2012_3/a4-3.pdf . (In Ukrainian)
8. Lavrischeva E., Ostrovski A., Radetskyi I. Approach to E-Learning Fundamentals Aspects of Software Engineering //8– th international Conf. ICTERI– 2012 “ICT in Education, Research and Industrial Applications”.– Publ. springer –sbm.com/ocs/home/ ICTERI–2012, Kherson, Ukraine, June 6–10, 2012. – Available at http://ceur-ws.org/Vol-848/ICTERI-2012_CEUR-WS-paper-17-p-176-187.pdf .
9. Kolesnyk A., Slabospitskaya O. Tested Approach for Variability Management Enhancing in Software Product Line. – Там же, CEUR–WS.org/Vol–848, ISSN 1613–0073, urn: nbn: de: 0074–848–8. – P. 125–133. – Available at http://ceur-ws.org/Vol-848/ICTERI-2012_CEUR-WS-paper-31-p-155-162.pdf
10. Lavrischeva E., Dzubenku A., Aronov A. Conception of Programs factory for Representation and E-learning Disciplines of Software Engineering // 9– th International Conf. ICTERI– 2013 “ICT in Education, Research and Industrial Applications; Integration, Harmonization and Knowledge Transfer”, Ukraine, June 17–21, 2013 <http://ceur-ws.org/Vol-1000/>

Надійшла до редколегії 27.06.13

УДК 04.04

¹Лавріщева К.М., д.-р ф.-м. н., проф.

²Колесник А.Л. і ³Стеняшин А.Ю. аспіранти

Об'єктне-компонентне проектування (ОКП) програмних систем. Теоретичні і практичні питання

Об'єктне-орієнтоване програмування (ООП) і компонентне програмування (КП) як різновиди збирального програмування, де роль елементів зборки відіграють об'єкти, компоненти та інтерфейси. Проектування програмних систем (ПС) пропонується виконувати новими засобами ООП і КП. Удосконалено об'єктне-компонентний метод В.М.Грищенко у напрямку формалізації процесу проектування предметної області з об'єктів і інтерфейсів, перебудови їх методів компонентами, а також конфігурування об'єктів, нових компонентів та компонентів повторного використання (КПВ) у різні структури ПС і їх сімейств з забезпеченням взаємодії у сучасних середовищах. Підхід к проектуванню варіантних ПС включає об'єктне моделювання Про та формальне проектування компонентів. Математичний апарат ОКП – це моделі, методи проектування, алгебри операцій об'єктного і компонентного програмування та конфігурування об'єктів і КПВ в різні варіанти ПС. Деякі результати ОКП реалізовані в інструментально-технологічному комплексі (ІТК).

Ключові слова: об'єкт, reuses, теорія, моделі ОКП, інтерфейс, типи даних, програмна система, змінювання об'єкту, варіант, взаємодія.

¹Інститут програмних систем НАН України,
03680, м. Київ-187, просп. Ак. Глушкова 40
E-mail: lavryscheva@gmail.com

^{2,3}Київський національний університет імені
Тараса Шевченка, студенти

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

1. Вступ

На рубежі 80-х ХХ сторіччя Г.Буч запропонував об'єктний підхід, який змінив діючий процес програмування. В той час

¹K.M. Lavrischeva, Dr. Sci. (Phys.–Math.), prof.

²Kolesnuk A.L. and ³Stenyachin A.U., graduate students

Objective-component designing (OCP) of the program systems. Theoretical and practical questions

The Objective-Oriented programming (OOP) and Component programming (CP) is a variety of the assembling programming, where a role of elements of assembling is warmed by objects, components and interfaces. It is suggested to execute designing the program systems (PS) by the new facilities OOP and CP. The objective-component method V.M. Grischenko is improved at direction of formalization of designing process to the subject domain from objects and interfaces, alteration of their methods by components, and also configuring objects, new components and components of repeated use (CR) in the different structures PS, as family members with providing to their co-operation in the modern environments. Approach to the development of variant PS includes the objective designing and formal designing of components. Mathematical vehicle OCP – it is models, methods of designing, algebra of operations of the objective and component programming and configuring objects and the CR into the different variants PS. Some results OCP are realized in the instrumental-technological complex (ITC).

Key Words: object, reuses, theory, OOP models, interface, data types, program systems, and exchanging, variant, interoperability.

¹Institute of Software Systems NAS Ukraine,
03680, Kyiv-187, 40, Glushkov Ave.
E-mail: lavryscheva@gmail.com

^{2,3}Taras Shevchenko National University of Kyiv,
students

індустрію програмного забезпечення спіткала криза складності. Потрібна була не тільки теорія Буча, а і технологія виходу з кризи. Об'єктний підхід став узагальненням попереднього традиційного підходу до розробки програмного

забезпечення. І як сказав Б. Гейтс, "періодично з'являються технології-вбивці», які руйнують все». Об'єктно-орієнтована технологія не тільки змінила традиційний підхід з розроблення ПС процедурними мовами програмування (МП), а вона сформувала новий стиль програмування різних систем шляхом моделювання предметних областей (ПрО) об'єктами і їх методами. Появились об'єктне-орієнтовані мови програмування (ООМ), бібліотеки класів різних об'єктів, routines і типів даних. Ці елементи стали ресурсом загального призначення для розробки багатьох систем. Крім того, були створені системи автоматизації ООП (COM, Corba, DCE RPC тощо). Головне їх призначення – формальне визначення об'єктної моделі (ОМ) і брокер об'єктних запитів об'єктів. ПС, створені з об'єктів і елементів бібліотек знижують складність ПС, забезпечують поповнення і віддалення об'єктів без деяких труднощів. Тобто зменшується криза складності ПС.

Відділ «Програмна інженерія» ІПС НАНУ починаючи з 1992–2012 рр. замовляв фундаментальні проекти в ДКНТ і НАНУ з розвитку об'єктно-орієнтованого і компонентного програмування в напрямку вдосконалення процесів моделювання ПрО функціональними і компонентними об'єктами. В результаті було розроблено теорію і технологію ОКМ, яку сформулювали проф. Лавріщева К.М. і Грищенко В.М. в [1–9] і в його докторській дисертації. Зараз ОКМ отримав подальший розвиток на шляху створення теорії взаємодії і варіабельності ПС з урахуванням сучасних умов функціонування середовищ.

В даній роботі співавтори – аспіранти додали нові риси ОКМ [1, 2], реалізували операції щодо забезпечення варіантів ПС і взаємодії об'єктів між собою в просторі сучасних середовищ. Розроблено технологію побудови ПС, в якій об'єднані на однієї концептуальній основі теоретичний апарат об'єктного аналізу з описом об'єктів-функцій сучасними ООМ і подання їх програмними компонентами. В плані підтримки КП розроблено низку спрощених ліній, а саме такі – розробка компонентів повторного використання (КПВ), збирання компонентів та забезпечення їх взаємодії з залученням апарату перетворення типів даних загального значення до необхідних форматів готових і змінюваних об'єктів ПС [13]. Лінії реалізовані в ІТК за

участю співавторів та студентів кафедри ТТП і ІС факультету кібернетики [14, 2].

Далі розглядаються теоретичні та прикладні аспекти ОКП. Подано опис базових понять і положень удосконаленої теорії моделювання ПрО з об'єктів, доказу ізоморфізму відображення методів об'єктів компонентами та їх адаптації в різні середовища виконання.

2. Математичне моделювання об'єктної моделі ПрО

Об'єктну теорію побудовано з використанням базових понять об'єктного підходу Г.Буча та трикутника Фреге, виходячи з принципів:

- загальності об'єктного визначення, всі сутності – суть об'єкти;
- об'єктних відмінностей, кожний об'єкт – унікальний елемент;
- об'єктної впорядкованості, всі об'єкти впорядковані відповідно їх відношенням;
- цілісності ОМ, об'єкти і відношення між ними однозначно визначаються в ній на певному рівні абстракції і опису.
- інтероперабельності об'єктів за допомогою операцій виклику об'єктів на множині вхідних і вихідних інтерфейсів.

Г. Буч ввів в програмування поняття об'єкту, класу, спадкоємності, інкапсуляції, поліморфізму. Ці поняття увійшли до нових ООМ, що з'явилися на початку 90–х і зараз широко використовуються системно і практично.

Класом називається загальна властивість, яка притаманна сукупності об'єктів.

Методом (або функцією) називається операція над об'єктами деякого класу.

Під *спадкоємством* розуміється властивість довільного об'єкту зберігати поведінку базового (батьківського) об'єкту на застосовність цієї властивості або методу.

Інкапсуляція – доступність до об'єкту за його властивостями і методами виконувати сумісне зберігання даних і функцій з заховуванням внутрішньої інформації і особливостей реалізації.

Поліморфізм – можливість оперування об'єктами без однозначної ідентифікації їх типів.

Принципи, поняття ООП і використаний формалізм трикутника Фреге, дозволили дати узагальнене поняття об'єкту з денотатом, концептом і знаком (рис. 1) та відношеннями між ними.

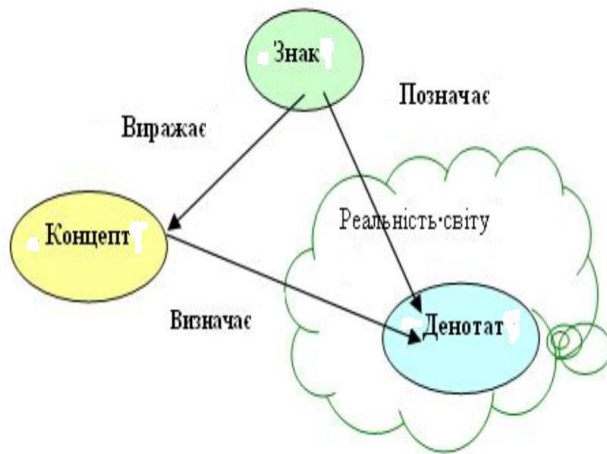


Рис.1. Трикутник Фреге об'єкту

Об'єкт виділяється на рівнях об'єктного аналізу із залученням математичних формалізмів для опису та уточнення функцій об'єктів у об'єктної моделі (ОМ), що створюється. ОМ відображає поняття і сутності ПрО, їх взаємовідношення та поведінку.

Об'єктна модель: «об'єктна орієнтація = об'єкти + наслідування, поліморфізм, інкапсуляція», класи об'єктів і відношення (агрегації, асоціації, спеціалізації, екземплярзації др.).

Побудова моделі ПрО починається з декомпозиції для виявлення об'єктів, визначення їх функцій і властивостей засобами логіко-математичного її моделювання та предикатами певної сигнатури [2, 8, 9].

Об'єкт ПрО – іменована частина дійсної реальності з певним рівнем абстракції відносно неї є понятійною структурою за трикутником Фреге (денотат, знак, концепт). Кожний об'єкт (О) як сутність множини об'єктів ПрО $O=(O_1, O_2, \dots, O_n)$, де $O_i = O_i(Na_i, Den_i, Con_i)$ і Na_i, Den_i, Con_i відповідно – знак (ім'я), денотат та концепт об'єкта. $Con_i = (Pi_1, Pi_2, \dots, Pi_s)$ визначається на множині предикатів $P=(P_1, P_2, \dots, P_r)$ [1, 2].

Аксіома 1. Предметна область, що моделюється з об'єктів, сама є об'єктом.

Аксіома 2. Предметна область, що моделюється, може бути окремим об'єктом у складі іншої предметної області.

При моделюванні об'єкт ПрО принаймні отримує хоча б одну властивість або характеристику, семантику та унікальну ідентифікацію в множині об'єктів ПрО та множині предикатів властивостей і відношень між об'єктами.

Властивість об'єкту визначається на множині об'єктів ПрО унарним предикатом, що приймає значення істини за його зовнішніми і внутрішніми властивостями. Множина предикатів моделі ПрО може бути довільною і визначається на відношеннях об'єктів.

Характеристика – це сукупність властивостей (унарних предикатів) за умовою приймання значень істини не більш, ніж одним предикатом з сукупності характеристик зовнішнього і внутрішнього типу із області значень властивості, що має істину.

Відношення завдається бінарним предикатом на множині об'єктів ПрО, приймає значення істини на поданій парі об'єктів. До основних типів взаємовідношень відносяться:

- 1) множина – множина;
- 2) елемент множини – елемент множини;
- 3) елемент множини – множина;
- 4) множина – елемент множини.

Цим типам відношень відповідають операції: *узагальнення, спеціалізації, агрегації, асоціації, деталізації, класифікації та екземплярзації*. Тип 3), 4) має *роде-видове* відношення (*IS-A*) та відношення *частина-ціле* (*PART-OF*).

Підхід до проектування моделі ПрО

Проектування ПрО виконується за чотирма рівнями [1-4]:

- *узагальнюючий* для визначення базових понять ПрО без урахування їх сутності та властивостей;
- *структурний* для впорядкування об'єктів в структурі моделі ОМ ПрО з відношеннями між ними;
- *характеристичний* для формування концептів об'єктів на базі їх властивостей та характеристик;
- *поведінковий* для визначення поведінки об'єктів в залежності від подій, що створюють об'єкти.

У відповідності з *узагальнюючим* рівнем об'єкт розглядається як математичне поняття, як клас, який в деякому випадку розглядається з точки зору аксіоматичної теорії множин Геделя-Бернайса: $O=(O_0, O_1, \dots, O_n)$, де O_0 – об'єкт ПрО.

На цьому рівні формується множина базових функцій ПрО, яка пов'язана з декомпозиційними або композиційними змінами денотатів та концептів об'єктів. Вона складається з ряду функцій, які охоплюють трансформації денотатів

та концептів у процесі об'єктного аналізу і до складу яких входять зміни, що пов'язані зі збільшенням або зменшенням кількості об'єктів (деталізація, екземплярізація, агрегація та ін.) та розширення або звужування концептів об'єктів. Проте ці зміни підпорядковуються певними правилами та умовами і забезпечують коректність виконання функцій.

Над множинами об'єктів можуть виконуватися базові операції над множинами (об'єднання, перетину, різниці, доповнення, приналежності тощо). Для множини об'єктів $O=(O_0, O_1, \dots, O_n)$ виконується відношення:

$$\forall i[(i>0) \& (O_i \in O_0)]. \quad (1.1)$$

За **структурним** рівнем визначаються такі поняття, як клас, екземпляр класу, абстрактний клас та ін. Визначення термінів властивостей об'єктів та їх відношень (агрегація, деталізація, класифікація тощо) виконується на характеристичному рівні, коли визначаються терміни: атрибут стану, стан, простір станів, метод та ін. Множина об'єктів упорядковується і кожен з об'єктів може подаватися як множина або елемент певної множини. Тоді вираз (1.1) трансформується у такий вигляд

$$\forall i \exists j[(i>0) \& (j \geq 0) \& (i \neq j) \& (O_i \in O_j)] \quad (1.2)$$

Він визначає відношення "частина-ціле", екземплярізація і агрегація.

У відповідності з **характеристичним** рівнем для кожного з об'єктів формується відповідний концепт. Якщо $O'=(O_1, O_2, \dots, O_n)$ – множина об'єктів ПрО, а $P'=(P_1, P_2, \dots, P_r)$ – множина унарних предикатів, що пов'язані із властивостями об'єктів ПрО, то концепт об'єкту O_i є множиною тверджень, побудованих на основі предикатів з P' , що приймають значення істини для об'єкту. Тобто концепт $Con_i = \{P_{ik}\}$ за умови $P_k(O_i) = true$, де P_{ik} є твердженням для об'єкту O_i відповідно предикату P_k . Згідно цих правил визначаються властивості об'єктів в рамках даної абстракції з відношенням типу "рід-вид".

Вираз $A = (O', P')$ визначає алгебраїчну систему концептів об'єктів O' і предикатів P' . Предикати P' містять операції:

0-арні операції, що відповідають константам;

унарні операції, що відповідають властивостям об'єктів;

бінарні операції, що відповідають взаємозв'язкам між парами об'єктів.

Предикатам відповідають умови:

– кількості предикатів достатньо для концептуального моделювання Про із множини об'єктів ПрО;

– кожен предикат, його тип та сигнатура повинні відображати сутність об'єкту.

Аксиома 1. Кожний об'єкт ПрО принаймні має хоча б одну властивість або характеристику, котра визначає семантику і унікальну ідентифікацію в множині об'єктів ПрО.

Згідно **поведінкового** рівня визначається послідовність станів об'єктів та їхні процеси з відображенням переходів станів. Взаємозв'язки між об'єктами формуються на основі бінарних предикатів, які пов'язані з властивостями об'єктів ПрО, і деталізуються до рівня взаємозв'язків між станами об'єктів.

Поняття класу об'єктів замінюється поняттям множини. Якщо об'єкт є елементом іншого об'єкту, то він визначається множиною. При цьому не кожен об'єкт є елементом будь якого іншого об'єкту (класу). Наприклад, об'єкт, що відповідає усій ПрО для певної ОМ, не є елементом будь якого іншого об'єкту у цієї ОМ. Визначення об'єкту формулюється за умовою: кожен об'єкт обов'язково є множиною або елементом деякої множини.

Впорядкування об'єкту виконується з урахуванням відношення належності, впорядкування елементів множини через числову множину (наприклад, множина природних чисел). Об'єкт вирізняється з множини усіх елементів завдяки побудованого відображення.

Конкретизацією поняття об'єкту у даній концепції є *клас* – це об'єкт, який подає собою множину; *екземпляр класу* – об'єкт, що є елементом певної множини, яка сама є класом; *об'єднаний клас* – множина, що є прямою сумою декількох інших множин; *клас-перехрещення* – це множина, яка є загальною частиною інших множин; *агрегований клас* – це множина, яка є підмножиною певного декартового добутку декількох інших множин.

Алгебра об'єктного аналізу ПрО

Ця алгебра має вигляд:

$$\Sigma = (O', I', A', P'), \quad (1.3)$$

де $O'=(O_1, O_2, \dots, O_n)$ – множина об'єктів, $I'=(I_1, I_2, \dots, I_n)$ – множина інтерфейсів O' ; $A'=(A_1, A_2, \dots, A_n)$ – множина (Action – A') операцій над елементами множини O ; $P'=(P_1, P_2, \dots, P_r)$ – множина предикатів з визначення властивостей концептів об'єктів. Кожна з операцій A' має певний пріоритет та арність, а також пов'язана з

відповідними допустимими описами концептів об'єктів та операцій множини $A' = \{decds, decdn, comds, comdn, conexp, connar\}$. Тобто *decds*, *decdn* – декомпозиції, *comds*, *comdn* – композиції та *conexp*, *connar* – звуження [1-4].

Теорема 1. Множина операцій A' алгебри об'єктів є системою дій щодо функцій об'єктного чотирьох рівневого подання моделі ПрО та операцій деталізації, екземпляризації і агрегації.

Операціями об'єктного аналізу є:

- конкретизація об'єкту, як класу, екземпляр класу тощо;
- аналіз сутностей: *0-арні*, *унарні*, *бінарні*.
- взаємовідношення *узагальнення*, *спеціалізація*, *агрегація*, *деталізація*, *класифікація*, *екземпляризація*;
- операції поведінки об'єктів з урахуванням зв'язків між характеристиками та часу їх існування в ОМ.

Модель ПрО подається об'єктним графом $G = \{O, I, R\}$, визначеним на множині об'єктів O , інтерфейсів I та відношень R (relations) між об'єктами:

- множина вершин O подає взаємно однозначність відображення об'єктів ПрО;
- кожній вершині відповідає хоча б один інтерфейс $I_k \in I$ та відношення, що належать множині R відповідно правил:
- існує хоча б одна вершина, що має статус множина-об'єкт і відображає ПрО в цілому.

Множина об'єктів-функцій O пов'язана з набором методів реалізації віддалених об'єктів ПрО. При конкретизації об'єкти G мають зв'язки через інтерфейсні об'єкти множини I . Тобто вершинами графу G є об'єкти двох типів – функціональні O та інтерфейсні I (рис.2).

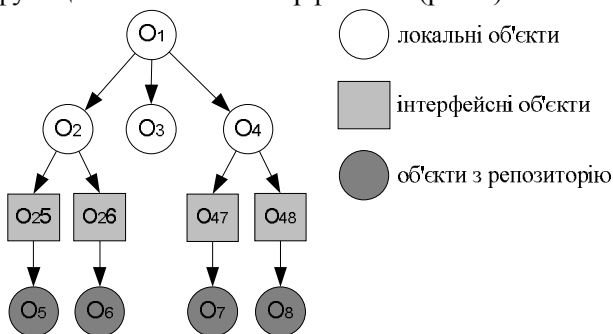


Рис.2. Приклад об'єктно-інтерфейсного графу

Інтерфейсним об'єктам графу відповідає опис даних, методи їх передачі через запити RPC чи RMI даних та можливі операції перетворення цих даних до відповідних форматів середовища виконання.

Результатом зв'язку двох об'єктів графу (наприклад, O_{25} , O_{47}) є інтерфейсний об'єкт, у якого множина вхідних інтерфейсів співпадає з множиною інтерфейсів об'єкта-приймальника, а множина вихідних інтерфейсів з множиною вихідних інтерфейсів об'єкта-передавача.

Аксиома 2. Побудований граф G доповнений інтерфейсними об'єктами структурно впорядкований (нагору) з контролем повноти і надмірності і усунення дублюючих елементів.

Об'єкти можуть мати декілька інтерфейсів, які можуть успадковувати інтерфейси інших об'єктів, тоді останні надають сервіс всієї множини вихідних інтерфейсів.

Множина об'єктів і інтерфейсів графу подається загальними чи індивідуальними властивостями та характеристиками в об'єктній моделі. Перевірка властивостей об'єктів проводиться операціями (екземпляризація, класифікація, спеціалізація, агрегація та ін.). Кожна з них це попарне порівняння внутрішніх характеристик об'єкту з їх зовнішніми характеристиками. Вони вважаються достовірними, якщо виконанні такі умови: кожна внутрішня властивість еквівалентна зовнішній властивості об'єкту. Якщо ця умова не виконується, то такий елемент віддалиться зі списку елементів множини і з графа відповідно.

Інтерфейси об'єктів

Вивченням проблеми інтерфейсів займався другий аспірант і співавтор [10]. Ним розроблені операції оброблення інтерфейсів, включаючи опис вхідних (*In*) і вихідних (*Out*) параметрів. Множина інтерфейсів $I = (In(O_k))$, необхідна іншому взаємодіючому з ним об'єктом, а множина вихідних інтерфейсів $Out(O_k)$ поверненню отриманих результатів.

Визначення семантики інтерфейсних об'єктів графу G виконується спеціальною мовою IDL з опису інтерфейсу (параметри *In*, *Out*), що запропонована групою OMG, що вперше реалізувала ОМ в системі CORBA та методику конструювання розподілених об'єктів [5-12]. Розроблені операції приналежності інтерфейсів:

$O_k \in O, In(O_k)$ – множина вхідних (*In*) інтерфейсних об'єктів;

$O_k \in O, Out(O_k)$ – множина вихідних (*Out*) інтерфейсних об'єктів.

Результатом взаємодії двох об'єктів з графу G (рис. 2) буде проміжний об'єкт, у якого множина вхідних інтерфейсів співпадає з множиною вхідних інтерфейсів об'єкта-

приймальника, а множина вихідних інтерфейсів з множиною вихідних інтерфейсів об'єкта-передавача:

$$\begin{aligned} O_k &= (Out(O_k), In(O_k)), \\ O_l &= (Out(O_l), In(O_l)), \\ O_k \cdot O_l &= (Out(O_k), In(O_l)) \end{aligned} \quad (1.4)$$

Композиція об'єктів $O_k \cdot O_l$ є коректною, якщо об'єкт-передавач забезпечує необхідний сервіс об'єкту-приймальнику:

$$\forall I_m \in In(O_k) \Rightarrow \exists I_n \in Out(O_l) \wedge I_m = I_n.$$

Об'єкти можуть мати декілька інтерфейсів, які можуть успадковувати інтерфейси інших об'єктів ($O_k \leftarrow O_l$), тоді останні надають сервіс всієї множини вихідних інтерфейсів: $O_k \leftarrow O_l \Rightarrow Out(O_k) \subseteq Out(O_l)$.

У випадку, коли об'єкт успадковує інший об'єкт, у якого множина вихідних інтерфейсів містить всі його інтерфейси, а множина вхідних інтерфейсів містить тільки інтерфейси, необхідні для надання сервісу.

Успадкований об'єкт делегує усі інтерфейси і має властивості:

транзитивності

$$\forall O_{1,2,3} \in O : O_1 \leftarrow O_2, O_2 \leftarrow O_3 \Rightarrow O_1 \leftarrow O_3$$

рефлексивності $\forall O_k \in O \Rightarrow O_k \leftarrow O_k$.

Результатом відображення об'єкту на об'єкт є інтерфейс з множини вхідних інтерфейсів $O_k[O_l] = O_k[In(O_l)]$ та вихідних інтерфейсів $O_k[O_l] = Ok[Out(O_l)]$.

Мова опису інтерфейсів подано в БНФ. Вона відображає фрагмент мови IDL з опису інтерфейсу [6].

```
<інтерфейс об'єкта> ::= Object <ім'я об'єкту> :{ <множина вихідних інтерфейсів>;
{<множина вхідних інтерфейсів>} end
<множина вхідних інтерфейсів> ::=
<множина інтерфейсів>,
<множина вихідних інтерфейсів> ::=
<множина інтерфейсів>,
<множина інтерфейсів> ::=  $\emptyset$  | <інтерфейс>;
<множина інтерфейсів>;
<інтерфейс> ::= Interface <ім'я_інтерфейсу>
: {<множина функцій>} end
<множина функцій> ::=  $\emptyset$  | <функція>;
<множина функцій>;
<функція> ::= function <ім'я_функції> :
<множина параметрів> end
<множина параметрів> ::= <параметр> |
<параметр>, <множина параметрів>
```

<параметр> ::= <тип> (<вид параметру>)

<вид параметру> ::= **in** | **out** | **inout**

Тип описує множину типів даних, що передаються через ці параметри операторами RPC, RMI, серісними протоколами WCF VS.Net чи Інтернет. Типи даних об'єктів описуються засобами OOM (C#, Vbasic, Pascal, та ін.) звичайних МП. Параметри об'єкту мають вид: **in** — вхідний параметр, **out** — вихідний параметр, **inout** — сумісний параметр.

Приклад опису параметрів інтерфейсів в програмах посередників типа stub F_1 і skeleton F_2 наведений нижче і використані в схемі (рис. 3) стосовно системи Corba [5].

```
interface  $F_1$  {
void f (in float s [1]);
}
```

```
interface  $F_2$  {
const long l=3 }
```

```
interface  $P_3$ :  $F_1()$ ,  $F_2()$ .
```

Вхідні та вихідні інтерфейси мають різну семантику, але однаковий синтаксичний опис. На формальному рівні інтерфейс може визивати перебудову типів даних переданих від **in** параметрів до **out** і зворотно.

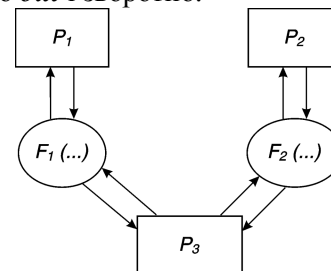


Рис.3. Схема виклику функцій об'єктів

Класи об'єктів. Об'єкти об'єднуються відповідно загальних характеристик у класи, тобто ОМ має такий вигляд:

$$OM = (Oclass, GK),$$

де $Oclass = \{Oclass^i\}$ — множина класів об'єктів функцій чи методів з загальними властивостями; GK — об'єктний граф, що відбиває зв'язки і відношення між класами й її екземплярами.

Кожен клас представляється у вигляді:

$$Oclass^i = (ClassName^i, Meth^i, Field^i), \quad (1.5)$$

де $ClassName^i$ — ім'я класу; $Meth^i = \{Meth_j^i\}$ — множина методів; $Field^i = \{Field_n^i\}$ — множина перемінних, що визначають стан екземплярів класу.

Нехай $Pfield^i \subset Field^i$ — множина зовнішніх перемінних (public), що доступні поза. Кожному $Pfield_n^i \in Pfield^i$ відповідає метод $get \langle Pfield_n^i \rangle$ і $set \langle Pfield_n^i \rangle$ для завдання і вибірки значень

відповідних перемінних, як атрибутів об'єктів і інтерфейсів ОМ і компонентних моделей, в яких методи об'єктів відображаються програмними компонентами. Множина методів має вигляд:

$$Imeth^i = Meth^i \cup \{get \langle Pfield_n^i \rangle\} \cup \{se \langle Pfield_n^i \rangle\}.$$

Йому зіставляється інтерфейс $Ifunc^i$, який складається з методів, що входять у $Imeth^i$.

Приклад класів (клієнт і сервер) наведено в комплексі ІТК (<http://sestudy.edu-ua.net>) двома мовами Java і C# в розділі ІТК «Зборка».

Клас – *клієнт* містить механізми комунікації між програмами, які генеруються для передачі даних до серверу, як екземплярів класу через

```
int func (int param);
```

```
int param;
```

```
...
```

```
ClientInterface ci = new ClientInterface();
```

```
int result = ci.func(param); ...
```

Клас-*сервер* генерує екземпляр класу для зворотній передачі даних клієнту:

```
class ServerInterface {
```

```
...
```

```
public int func (int param){
```

```
...//
```

```
void main(){
```

```
...
```

```
ServerInterface si = new ServerInterface();
```

```
si.work();
```

```
...}
```

Опис програми взаємодії *ClientInterface* і *ServerInterface* і її реалізація наведена в ІТК в тому ж розділі.

Сучасні інструменти реалізації об'єктів

1. **Component Object Model** (COM) для обробки компонент в Visual Basic, C++, .NET і ін.
2. **Java Beans** – стандарт Sun Microsystems для компонентів незалежно від МП.
3. **CORBA** – IDL-інтерфейс, об'єкти, моделювання варіантів об'єктів і ОМ, як в системі COM.
- 4) **Microsoft**, Sun Microsystems і JavaBeans використовують підмножину типів даних у всіх мовах ООМ системи .NET, а саме: CLS (Common Language Specification), загальних програм CLR (Common Language Runtime), система типів CTS (*Common Type System*), що включає опис типів даних в мові специфікації CLS для відображення типів даних МП в систему типів .NET.
- 5) **IBM, COBRA** забезпечують розробку ПС з компонентів і сервісів через middleware, за допомогою якої виконується адаптація виконання системи у їх середовищі.

3. Компонентне проектування ПС

Reuseability - повторне використання КПВ є базисом сучасних індустріальних технологій і фабрик програм. Їй приділяється велика увага за кордоном. Вона досліджується у державних програмах Японії, Великобританії, США й ін., а також у світових корпораціях (IBM, Microsoft, Hewlett-Packard і ін.). Проблема reuses розглядаються на міжнародних конференціях протягом десятиріч. В reuses вкладаються великі кошти для одержання прибутку при розробленні з них складних ПС [8, 9, 11, 12].

Основні поняття

Під *програмним компонентом* чи просто компонентом розуміється незалежний від МП, самостійно програмний об'єкт, який забезпечує виконання певної сукупності прикладних задач, доступ до яких можливий тільки за інтерфейсами, які задають параметри і операції звернення до них.

КПВ як елемент типового повторного рішення в ПС потрібен мати типову архітектуру, характеристики і атрибути, які подаються в інтерфейсній частині і застосовуються при обміні даними для взаємодії компонентів у різних середовищах. Тобто поданий таким чином компонент, стає неподільним та інкапсульованим об'єктом, який задовольняє функціональним вимогам, а також вимогам щодо архітектури системи і середовища виконання компонентній програми.

Компонентна програма чи *ПС* із компонентів – це сукупність компонентів, що реалізують функціональні та не функціональні вимоги до неї, будується за правилами конфігурацій збирального типу з забезпеченням взаємодії.

Об'єктна модель ПрО є джерелом трансформаційного переходу від об'єктів до компонентів з використанням математичного апарату моделювання: моделей компонента і інтерфейсу, компонентного середовища, зовнішній і внутрішній компонентної алгебри та алгебраїчної системи перетворення типів даних компонентів для їх взаємодії [9]. Побудова компонентних ПС з готових КПВ із різних бібліотек дає значне скорочення витрат і поліпшує якість продукту [8, 14, 15].

Модель компонента є наслідком узагальнення типових рішень і має вигляд:

$$C = (CNa, CIn, CFa, CIm, CSe), \quad (2.1)$$

де CNa – унікальне ім'я компонента, $CIn = \{CIn^i\}$ – множина інтерфейсів, пов'язаних з компонентом, Cfa – інтерфейс забезпечення функцій керування екземплярами компонента, $CIm = \{CIm^i\}$ – множина реалізацій компонента, $CSe = \{CSe^i\}$ – множина загальносистемних сервісів.

Множина інтерфейсів $CIn = \{CInIn \cup CInOut\}$ складається з вхідних $CInIn$ та вихідних $CInOut$ параметрів інтерфейсу. Тобто компонент має вхідні інтерфейси при власній реалізації, а вихідні інтерфейси реалізації іншого компонента. Кожен з них має відповідну модель

$$CIn^i = (InNa^i, InFu^i, InSp^i), \quad (2.2)$$

де $InNa^i$ – ім'я інтерфейсу, $InFu^i$ – функціональність (сукупність методів), $InSp^i$ – специфікація інтерфейсу: опис типів, констант, інших елементів сигнатур методів, характеристик тощо.

Інтерфейс функції Cfa визначає методи і звернення до екземплярів компонентів (пошук, вибір, знищення тощо).

Кожна реалізація $CIm^i \in CIm$ задається моделлю:

$$CIm^i = (ImNa^i, ImFu^i, ImSp^i) \quad (2.3)$$

де $ImNa^i$ – ідентифікатор або ім'я реалізації, $ImFu^i$ – функціональність, $ImSp^i$ – специфікація опису (умови виконання, залежності від певних платформ тощо).

Необхідною вимогою існування компонента є умова його цілісності:

$$\forall CIn^i \in CInI \exists CIm^i \in CIm [Pr(CIn^i) \subseteq CIm^i], \quad (2.4)$$

де $Pr(CIn^i)$ означає функціональність з реалізації методів інтерфейсу CIn^i .

Аксіома 3. Для об'єднання двох різнорідних компонентів C_1 та C_2 існують умови, якщо $CIn_1^i \in CInO_1$ і $CIn_2^k \in CInI_2$ такий, що

$$S(CIn_1^i) = S(CIn_2^k) \& Pr(CIn_1^i) \subseteq CIm_2^i,$$

де $S(sign)...$ означає сигнатуру відповідного інтерфейсу.

Модель компонентної ПС. Ця модель ПС (PS) ідентична моделі програмного компоненту з той різниці, що її елементами можуть бути самі програми, як самостійні компоненти.

$$PS = \{PSLm\{Lm^1, \dots, Lm^n\}, R\{Ri, \dots, Rm\},$$

$$PSLn\{Ln^1, \dots, Ln^k\},$$

де $PSLm\{Lm^1, \dots, Lm^n\}$ – множина реалізацій компоненту, КПВ, ПС;

$$R\{Ri, \dots, Rm\} - \text{множина предикатів};$$

$$PSLn\{Ln^1, \dots, Ln^k\} - \text{множина інтерфейсів компонентів і програм.}$$

Операції множини R можуть відповідати об'єднанню чи конфігурації КПВ в деякі складні

структури програм і ПС з множини компонентів, інтерфейсів.

Компоненти чи КПВ можуть змінюватися, замінюватися новими функціонально подібними, еквівалентними чи тотожними КПВ з метою отримання нових варіантів продукту ПП за аксіомами.

Аксіома 3. Два компоненти C_1 та C_2 є тотожними (рівними), якщо тотожними є їх відповідні складові. Як наслідок, заміна C_1 на C_2 не впливає на компонентну програму, до якої належить C_1 .

Аксіома 4. Два компоненти C_1 та C_2 є еквівалентними, якщо тотожними є їх множини інтерфейсів та реалізацій. Заміна C_1 на C_2 не змінює функціональності компонентної програми за умови встановлення відповідності між іменами у самій програмі.

Аксіома 5. Два компоненти C_1 та C_2 є подібними, якщо тотожними є їх множини інтерфейсів. Заміна C_1 на C_2 зберігає взаємозв'язки компонентів, але функціональність компонентної програми може змінитись.

Головна мета КП це зв'язування (зборка) компонентів, а також змінювання деяких КПВ тотожними чи еквівалентними в структурі ПС. Зв'язування має свою арність і може складатися з множини даних, які входять до класу інтерфейсів означеного класу КПВ. Ця операція виконується за моделлю компоненту (2.1) і інтерфейсу (2.2), які містять предикати, які визначають умову передачі даних іншим компонентам.

Між компонентами можуть існувати відношення: успадкування, екземпляризації, контракту, об'єднання (зв'язування). Усі відношення визначаються на інтерфейсах і предикатах і мають такий зміст.

Відношення екземпляризації. Вираз $CIns_k^{ij} = (Ins_k^{ij}, InFu^i, ImFu^j)$ описує певний екземпляр компонента C , де: Ins_k^{ij} – унікальний ідентифікатор екземпляра, $InFu^i$ – функціональність інтерфейсу $CIn^i \in CIn$, $ImFu^j$ – програмний елемент, що забезпечує виконання реалізації $CIm^j \in CIm$.

Відношення контракту. Між компонентами C_1 і C_2 існує відношення контракту

$$Cont_{12}^{im} = (CInO_1^i, CInI_2^m, IMap_{12}^{im}),$$

де $CInO_1^i \in CIn_1$ – вихідний інтерфейс першого компонента, $CInI_2^m \in CIn_2$ – вхідний інтерфейс другого, $IMap_{12}^{im}$ – відображення відповідності між методами, які входять до складу обох інтерфейсів з врахуванням сигнатур та переданих

типів даних. Якщо компонент C_2 має реалізацію інтерфейсу $CInI_2^m$, то забезпечується виконання функціональності $InFu_1^i$ інтерфейсу $CInO_1^i$.

Відношення зв'язування. Якщо між компонентами C_1 і C_2 існує відношення контракту $Cont_{12}^{im}$, то між їх екземплярами $CIns_{1k}^{ij} = (InS_{1k}^{ij}, InFu_1^i, ImFu_1^j)$ та $CIns_{2p}^{mq} = (InS_{2p}^{mq}, InFu_2^m, ImFu_2^q)$ існує відношення зв'язування відносно контракту $Cont_{12}^{im}$, яке описується у вигляді

$$Bind\ InS_{1k}^{ij}, InS_{2p}^{mq}, Cont_{12}^{im}.$$

Відношення взаємодії. Якщо компоненти ПС розташовані у розподіленому середовищі і вони входять до множини компонентів репозиторію ПС, то використовується модель взаємодії, визначеної на множині інтерфейсів готових КПВ та віддалених викликів [11, 12].

Модель компонентного середовища має наступний вираз:

$$CE = (CNa, InRep, ImRep, CSe, CSeIm), \quad (2.4)$$

де $CNa = \{CNa^m\}$ – множина імен компонентів, які входять до складу середовища, $InRep = \{InRep^i\}$ – репозитарій інтерфейсів компонентів середовища, $ImRep = \{ImRep^j\}$ – репозитарій реалізацій, $CSe = \{CSe^r\}$ – інтерфейс системних сервісів, $CSeIm = \{CSeIm^r\}$ – множина реалізацій для системних сервісів.

Кожен елемент з $InRep$ це двійка (CIn^i, CNa^m) , де CIn^i – інтерфейс компонента, CNa^m – ім'я компонента, для якого існує інтерфейс. Аналогічно кожен елемент з $ImRep$ описується (CIm^j, CNa^m) , де CIm^j – реалізація, яка описується виразом (4), а CNa^m – ім'я компонента, до якого належить ця реалізація.

Компонентне середовище на рівні моделі розглядається як множина серверів застосувань, де розгортаються компоненти-контейнери, екземпляри яких реалізують функціональність компонента. Взаємозв'язок контейнера з сервером забезпечується через стандартизовані інтерфейси (CFa) . Зв'язки між КПВ, які розгорнуті у різних серверах забезпечуються реалізаціями інтерфейсу CSe .

До складу алгебри компонентів входять операції:

– інсталяції (розгортання компонента) $CE_2 = C \oplus CE_1$;

– об'єднання компонентних середовищ $CE_3 = CE_1 \cup CE_2$;

– видалення компонента з компонентного середовища $CE_2 = CE_1 \setminus C$.

Теорема 2. Кожне компонентне середовище CE є результатом виконання послідовності

операцій розгортання компонентів, які входять до його складу, у певному компонентному каркасі: $CE = C_1 \oplus C_2 \oplus \dots \oplus C_n \oplus FW$.

Теорема 3. Побудова компонентного середовища не залежить від порядку інсталяції компонентів, які входять до складу цього середовища, тобто:

$$C_1 \oplus (C_2 \oplus CE) = C_2 \oplus (C_1 \oplus CE), \quad (2.5)$$

Теорема 4. Операція об'єднання компонентних середовищ асоціативна:

$$(CE_1 \cup CE_2) \cup CE_3 = CE_1 \cup (CE_2 \cup CE_3), \quad (2.6)$$

Теорема 5. Операція об'єднання компонентних середовищ комутативна:

$$CE_1 \cup CE_2 = CE_2 \cup CE_1. \quad (2.7)$$

Теорема 6. В довільних компонентних середовищах CE_1 і CE_2 та компонента C завжди виконується:

$$C \oplus (CE_1 \cup CE_2) = (C \oplus CE_1) \cup CE_2 = (C \oplus CE_2) \cup CE_1. \quad (2.8)$$

Теорема 7. Для будь-якого компонента C та компонентного середовища CE завжди виконується: $(C \oplus CE) \setminus C = CE$.

Модель ПС для певного компонентного середовища описується виразом

$$CP = (CE, Cont, O),$$

де CE – компонентне середовище, $Cont$ – множина контрактів для компонентів, що входять до складу CE , O – підмножина компонентів з CE , що включають реалізації і звертаються до інших компонентів за допомогою своїх вихідних інтерфейсів.

Умова цілісності компонентної програми полягає в існуванні для кожного компонента C_1 з CE , що має вихідний інтерфейс $CInO_1^i$, компонента C_2 з відповідним вхідним інтерфейсом $CInI_2^m$ і контракт $Cont_{12}^{im} = (CInO_1^i, CInI_2^m, IMap_{12}^{im})$ входить до складу множини $Cont$.

Для реалізацій функціональності компонентів застосовуються різні мови програмування з відповідними типами даних в них. При збиранні (об'єднанні, інтеграції) різномовних компонентів вирішується задача перетворення типів даних переданих інтерфейсом для релевантної конвертації типів даних відповідно моделі інтегрованого середовища.

Існує два підходи при вирішенні проблеми інтеграції КПВ: 1) застосування моделі взаємодії пари різномовних компонентів в МП у середовищі розподілених систем шляхом їх композиції через модуль посередник (stub, skeleton) з використанням міжмовного і

міжмодульного інтерфейсу [11]; 2) опис інтерфейсу КПВ в мові IDL за параметрами *In*, *Out*, *Inout* для завдання даних при обміні даними між собою [9, 10].

В теоретичному аспекті інтеграція різномовних компонентів базується на формальних відображеннях, поданих як суперпозиції базових відображень.

Модель збирання різнорідних КПВ

Нехай задана множина компонентів $CSet = \{C_{ij}\}$ в різних МП. При взаємодії компоненти обмінюються даними, кожне з яких визначається трійкою: ім'ям змінної, її типом даних та значенням [9-12]. Дані, якими обмінюються кожна пара компонентів C_i і C_j , можуть бути еквівалентними, якщо вони мають однакову семантичну структуру і тип, або нееквівалентними і в цьому випадку необхідне їхнє перетворення за допомогою функцій, які подані відображеннями:

$FN_{ij}: N_i \rightarrow N_j, FT_{ij}: T_i \rightarrow T_j, FV_{ij}: V_i \rightarrow V_j$ (2.9), де FN_{ij} встановлюють відповідність між іменами змінних (наприклад, у множинах формальних та фактичних параметрів), FT_{ij} описують еквівалентні відображення для типів даних *in*, *out*, FV_{ij} реалізують необхідні перетворення значень інтерфейсних даних.

Задача заміни змінних FN_{ij} виконується шляхом впорядкування імен змінних (наприклад, в описі конфігурації компонентів і КПВ. Відображення між типами даних FT_{ij} базується на перетвореннях типів даних, кожен з яких подається як абстрактна алгебра або алгебраїчна система $T = (X, \Omega)$, де X – множина значень, що можуть приймати зміні цього типу, а Ω – множина операцій над цими змінними. Відображення FV_{ij} застосовуються у випадках нееквівалентності типів T_i та T_j (наприклад, перетворення цілого значення у дійсне) [14, 15].

У випадках багаторазових викликів методів компонентів можуть виникати питання прямого та оберненого перетворення даних між формальними та фактичними параметрами і тому є суттєва умова ізоморфізму відображення між відповідними типами даних. Під перетворенням типу $T_i = (X_i, \Omega_i)$ у тип $T_j = (X_j, \Omega_j)$ розуміється таке перетворення, при якому семантичний зміст операцій з Ω_i еквівалентний змісту операцій з Ω_j . У загальному випадку перетворення типу T_i у тип T_j може бути однобічним, коли еквівалентного перетворення не існує. Ці системи для МП поширені на загальні типи даних, що описані у

стандарті ISO/IEC 11404 і містять механізми генерації $GDT \leftrightarrow FDT$.

Задача взаємозв'язку пари компонентів (систем, ПС) виникає при їх збиранні у складні структури. Для цього побудовано сукупність відображень для різних видів викликів методів (Call, RPC, RMI), за якими встановлюються відповідність між множиною фактичних параметрів $V = \{v_1, v_2, \dots, v_k\}$ об'єкту і множиною формальних параметрів $F = \{f_1, f_2, \dots, f_l\}$.

Задачі побудови типів даних для основних типів даних, що застосовуються у різних МП і ізоморфні відображення між цими алгебраїчними системами детально розглянуто у [15].

При гетерогенному середовищі задача взаємодії вирішується шляхом:

1) перетворення типів даних, які застосовуються при передачі даних між взаємодіючими компонентами і системами;

2) усунення відмінностей у поданні даних за різними особливостями комп'ютерних платформ і архітектур, а також за принципами кодування (декодування) даних при передачі їх через мережу іншим системам;

3) описом інтерфейсу компонентів і систем мовою IDL, API для передачі даних через операції віддаленого виклику тощо.

Інтерфейсна модель з компонентів, КПВ і систем має вираз

$$ISyst = (IFu, IG), \quad (2.10)$$

де: $IFu = \{IFu^i\}$ – множина інтерфейсів, IG – інтерфейсний граф, який еквівалентний графу G . Для класів визначаються умови подання їх як елементів множини інтерфейсів.

Теорема 8. Для кожної моделі системи PS , зовнішня взаємодія з класами *public*-методи та керованими змінними, існує єдине інтерфейсне подання $ISyst$ з еквівалентною функціональністю.

Ця теорема визначає умови існування еквівалентного відображення між об'єктним та інтерфейсним об'єктом.

На даний час проблема перетворення даних має місце у стандарті загальних типів даних ISO/IEC 11404–2007 GDT. Нами пропонуються механізми генерації GDT до FDT та перетворення їх за допомогою *примітивних функцій* спеціальних бібліотек системи VS.Net (CLR, CTS і CLS тощо). Кожний компонент має паспортні дані для забезпечення інтерфейсу взаємодіючих компонентів (рис. 4). Він використовується на фабриці програм <http://programsfactory.univ.kiev.ua>), що входить до складу ІТК.

Параметр	Опис
Name	Ім'я набору відомостей, що складає компонент. Має бути унікальним у сховищі мета даних ETICS
DisplayName	Відображуване дружнє ім'я набору відомостей компонента. Не потребує унікальності
Platform	Одна з платформ розробки
Description	Опис набору відомостей компонента
PackageName	Ім'я пакету, виробленого цим компонентом, якщо воно відрізняється від імені компонента
Homepage	Адреса домашньої сторінки компонента
Vendor	Власник компонента (організація, компанія, особа)
ModifyDate	Дата останньої модифікації
Download	Шлях до компонента
Download run	Шлях до виконуваного файлу компонента

Рис.4. Стандарт опису даних компоненту в Grid

Виходячи з формального опису дані, якими обмінюються кожна пара компонентів C_i і C_j , можуть бути еквівалентними, якщо вони мають однакову семантичну структуру і тип, інакше – не еквівалентними.

Змінювання компонентних ПС

Програмна система будується з КПВ шляхом їх збирання та необхідного змінювання інтерфейсів та реалізацій компонентів або їх додавання [15].

Зовнішня алгебра $\Psi = \{CSet, CSEt, \Omega\}$ завдає операції над компонентами і середовищами. Внутрішня алгебра має операції реорганізації вихідного коду: *рефакторинг* для змінювання внутрішньої структури КПВ чи інтерфейсу зі збереженням його поведінки і функціональності; *реінженерії*, як апарату внесення деяких змін в КПВ або перепрограмування логіки, уточнення функцій окремих КПВ для їх змінювання в ПС; *реверсної інженерії* шляхом встановлення первісної структури в МП і її реконструювання:

$$\Sigma = \{\Psi \cap \phi\} = \{CSet, CSEt, \Omega\} \cap \{CSet, CSEt, \Omega\},$$

де Ψ – зовнішня алгебра, ϕ – внутрішня алгебра.

Зовнішня алгебра: $\Psi = \{CSet, CSEt, \Omega\}$,

де $CSet$ – множина компонентів C , $CSEt$ – середовище E з множини компонентів C і інтерфейсів In . $\Omega = \{CE1, CE2, CE3, CE4\}$ – операції алгебри: $CE1$ – операції оброблення компонентів, $CE2$ – операції інсталяції, $CE3 = CE1 \cap CE2$ – операції зборки (об'єднання), $CE4 = CE1 \setminus C$ – операції видалення компонента C з середовища,

$C2 - CE2 = C2 \oplus (CE1 \setminus C1)$ – операції заміщення.

Алгебра еволюції КП

Важливе значення в еволюції ПС мають методи відбору КПВ із різних сховищ, їх трансформація, а також перебудова під потреби ПРО засобами реінжинірингу і рефакторінгу [19].

Методи еволюції поділяються на два види: методи, що змінюють функціональність та поведінку компонентів, і методи, які пов'язані з нефункціональними змінами. До перших належать зміни в інтерфейсах (зміна сигнатур інтерфейсів, додавання інтерфейсів) та реалізації (зміна алгоритмів та логіки, заміщення та додавання реалізацій) та ін. До другого виду належать зміни, що пов'язані з нефункціональними характеристиками ПС (надійності, ефективності, мобільності), мовами та платформами виконання. Методам еволюції відповідають побудовані зовнішні і внутрішні алгебри.

Внутрішня алгебра еволюції:

$$\phi = \{CSet, CSEt, \Omega\},$$

де $\Omega = \{Oref, ORei, ORev\}$ – сукупність операцій еволюції компонентів.

На основі семантики, умов та вимог виконання функцій рефакторінгу побудовано внутрішню алгебра рефакторінгу компонентів

$\Sigma^f = (CSet, Ref)$, де $CSet = \{C_n\}$ – множина компонентів, модель кожного з яких наведена виразом (2.6), а $Ref = \{AdOIm, AdNIm, RelIm, AdIn\}$ – сукупність операцій рефакторінгу,

де $AdOIm$ – додавання нової реалізації для існуючого інтерфейсу.

$NewC = AdOIm(OldC, NewCIm^s, NewCInO^s)$ має семантику, де $NewCIn = OldCIn \cup NewCInO^s$ – операція додавання вихідних інтерфейсів для нової реалізації, $NewCIm = OldCIm \cup \{NewCIm^s\}$ – додавання нової реалізації.

Аксиома 7. Для $\exists OldCIn^t \in OldCInI$ існує операція додавання вхідних інтерфейсів і відповідної функціональності.

Ця операція є асоціативною та комутативною і є умовою цілісності компонента.

Операція $AdNIm$ є додаванням нової реалізації, вхідний інтерфейс, якої не входить до складу множини інтерфейсів компонента:

$$NewC = AdNIm(OldC, NewCIm^s, NewCInO^s).$$

В ній $NewCIn = OldCIn \cup NewCInO^s$ – додає нові вихідні інтерфейси для нової реалізації, $NewCIm = OldCIm \cup \{NewCIm^s\}$ – додає нову реалізацію. Ці операції є асоціативними та комутативними. Цілісність компоненту зберігається.

Операція $ReplIm$ є заміщенням існуючої реалізації новою без зміни вхідного інтерфейсу: $NewC = ReplIm(OldC, NewCIm^s, NewCInO^s, OldCIm^r, OldCInO^r)$. В ній $NewCIn = OldCIn \cup \{NewCInO^s\} \setminus \{OldCInO^r\}$ – додавання вихідних інтерфейсів для нової реалізації і видалення вихідних інтерфейсів старої реалізації, $NewIm = OldIm \cup \{NewCIm^s\} \setminus \{OldCIm^r\}$ – додавання нової реалізації і видалення застарілої. Умовою виконання операції є незмінність функціональності усіх інтерфейсів, що пов'язані зі старою реалізацією, після заміщення її новою:

$$\begin{aligned} & \forall OldCIn^i \in OldCIn \\ & [(Pr(OldCIn^i) \subseteq OldCIm^r) \Rightarrow \\ & (Pr(OldCIn^i) \subseteq NewCIm^s)] \vee \exists OldCIm^i \\ & \in (OldCIm \setminus \{OldCIm^r\}) [Pr(OldCIn^i) \subseteq OldCIm^i]. \end{aligned}$$

Лема 1. Операція заміщення існуючої реалізації новою з визначеними вище умовами та семантикою зберігає цілісність компонента.

Операція $AddIn$ є додаванням нового вхідного інтерфейсу для існуючої реалізації: $NewC = AddIn(OldC, NewCIn^q)$. В ній $NewCIn = OldCIn \cup \{NewCIn^q\}$ – додавання нового вхідного інтерфейсу $\exists OldCIm^s \in OldCIm [Pr(NewCIn^q) \subseteq OldCIm^s]$ з відповідною функціональністю.

Лема 2. Операція додаванням нового вхідного інтерфейсу для існуючої реалізації з визначеними вище умовами та семантикою зберігає цілісність компонента.

Наведені операції є базовими для більш складних ПС. Наприклад, додавання реалізації разом з вхідним інтерфейсом визначаються наступною суперпозицією:

$$NewC = AddIn(AddNIm(OldC, NewCIm^s, NewCInO^s), NewCIn^q).$$

Сформульовано та доведено наступну теорему.

Теорема 9. Алгебра рефакторингу компонентів $\Sigma rf = (CSet, Refac)$ є достатньо та несуперечна.

Рефакторинг компонентів. Результатом операцій рефакторингу або суперпозицій операцій є певний компонент. Тому $CSet$ складається з КПВ і різних модифікацій компонентів як результатів виконання операцій рефакторингу. Наприклад, для певної компонентної моделі конфігурація, яка складається з двох компонентів в вхідним і вихідним інтерфейсом.

Внутрішня компонентна алгебра реінжинірингу компонентів $\Sigma^{re} = (CSet, Re)$ може бути побудована лише за умови порушення

цілісності подання компонентів. Крім операцій рефакторингу (Ref) до множини входять операції Re , які видаляють з компонента існуючий інтерфейс або змінюють його сигнатуру. Це порушує умову цілісності, бо інші компоненти, які звертаються до нього, не зможуть мати доступу до необхідної функціональності.

У зв'язку з цим розглянуто модель реінжинірингу компонентів $M_{Re} = (CSet, Re)$. У цьому випадку семантика операцій з Re може полягати у трансформації не тільки цільового компонента, а й інших. Наприклад, при зміні сигнатури вхідного інтерфейсу необхідна одночасна зміна інших компонентів, в яких існують звернення до методів цього інтерфейсу.

Аналогічним чином сформовано відповідну модель реверсної інженерії $M_{Res} = (CSet, Rev)$, при цьому $Re \subset Rev$. Особливість множини Rev полягає у тому, що вона не визначена повністю (кількість операцій є необмеженою), а лише за класифікацією операцій. Наприклад, зміни якісних характеристик КПВ можуть виконуватись довільним способом і відповідні операції складають сукупність операцій для зміни певного показника якості (за класифікаційною ознакою в множини Rev).

Таким чином, алгебра Σrf та моделі M_{Re} і M_{Res} складають формальний апарат внутрішній алгебри щодо моделей і методів еволюції компонентів.

4. Розвиток моделей взаємодії і варіабельності ПС

Починаючи з 2007 р. у відділі активно розвивалася теорія взаємодії і варіантності ПС з КПВ в межах фундаментального проекту з генерувального програмування та комплексна технологія реалізації ПС і сімейств ПС із КПВ в інтегрованому середовищі сучасних систем Eclipse, VS.Net, Corba [12, 13].

Варіабельність ПС і сімейств

Вона бере свій початок з задач створення різних варіантів програмного продукту (наприклад, теорія рефакторингу М.Фаулера, Product line SEI тощо). Нами в рамках фундаментального проекту визначено клас моделей і процес керування варіантними структурами ПС і СПС з використанням КПВ і точок варіантності [12, 13, 22]. Модель варіантної структури і конфігурування членів СПС із КПВ будується на основі операцій змінення, поповнення новими формальними моделями, еквівалентними за функціональністю

КПВ, які створюють підклас рефакторингу варіабельних членів сімейства і СПС.

Моделі варіабельності СПС є пара під моделей: $VM = \{SV, AV\}$,

де SV – під модель варіантності структури СПС;

AV – під модель варіабельності продуктів процесу розроблення СПС.

Моделі VM у процесі розроблення СПС забезпечує рівень змінності продуктів СПС, адекватно заданим умовам їх використання, а також скорочення термінів розроблення продуктів СПС [11–13].

Підмодель структури СПС має вигляд:

$$SV = \{Gt, \{GR\}, Con, Dep\},$$

де $Gt = (F, LF) \square$ – граф, вершини якого ($f \in Ft$) є унікальні ідентифікатори артефактів різного типу (вимоги, КПВ, ПС тощо), поєднаних зв'язками обов'язкового та варіантного підпорядкування (LFt); Con і Dep – предикати на декартовому добутку множин артефактів усіх типів предикатів, які подають обмеження й залежності між функціями та характеристиками СПС.

Ці моделі описані в [4, 13]. Їх розвиток з точки зору теорії КП виконано першим співавтором у дисертаційній роботі.

Розвиток теорія взаємодії КПВ і систем.

Модель взаємодії призначена для обміну інформацією між різними компонентами і системами при їх об'єднанні й обчисленні. В основі взаємодії лежить процес обміну повідомленнями (викликами, запитами, протоколами) між програмами і системами для сумісного рішення певної задачі. В повідомленнях задається інтерфейс, який специфікується мовою IDL або іншими (APL, SIDL тощо). На загальному рівні інтерфейс є механізмом забезпечення взаємодії (interconnection) різнорідних програм для сучасних середовищ, що забезпечують виконання програм і систем [12, 18].

Модель взаємодії визначає взаємозв'язок між КПВ і ПС через параметри, що передаються між ними. Вона включає систему понять і відношень, які подаються математичними засобами – абстрактна алгебра, теорія множин тощо. Модель M_{inter} має загальний вигляд:

$$M_{inter} = \{M_{pro}, M_{sys}, M_{env}\}, \quad (2.9)$$

де $M_{pro} = \{C, Int, Pr\}$ – модель програми, C – компонентний об'єкт, Int – інтерфейс, Pr – протокол для передачі даних;

$M_{sys} = \{PS, Int, Pr\}$ – модель програмної системи PS , Int – інтерфейс, Pr – протокол для передачі даних;

$M_{env} = \{Envir, Int, Pr\}$ – модель середовища, в якому Int, Pr містить сукупність зовнішніх інтерфейсів та викликів, що передають дані між програмами через мережу.

Базовими параметрами моделі взаємодії M_{inter} є програма, інтерфейс і протокол. По відношенню до моделі відкритих систем OSI, M_{inter} є моделлю верхнього рівня.

5. Реалізація технології компонентного проєктування ПС в ІТК

Деякі аспекти теорії КП реалізовані практично в ІТК, як комплексна технологія. В ній об'єднанні на однієї концептуальній основі теоретичний апарат моделювання Про за об'єктами та теорією компонентного програмування ПС із різномовних КПВ на основі компонентної алгебри і апарата алгебраїчних систем з еквівалентним перетворенням типів даних КПВ (reuses) до потрібних форматів середовища виконання.

Комплексну технологію ТП подано нижче спектром ліній за веб-сайтом ІТК [13–20] <http://sestudy.edu-ua.net>.

TECHNOLOGIES OF WEB-SITE

Technology of service repository of reuses
Technology of development of reuses
Technology of assembling of reuses
Technology of configuring of reuses
Technology of generation of DSL
Technology of expenses quality
Technology of ontology LC ISO/IEC-12207
Technology of web-services
Technology of GDT ISO/IEC-11404 .

INTEROPEABILITY of the programs:

– Model of Corba – Eclipse – Java
– Model of VS.Net C#– Eclipse
– Model of Basic – C#.

TOOLS ITC: Eclipse, Protégé. VS.Net, Java PRESENTATIONS of science article:

Factories of the programs
Method of construction TL
Variability

E-LEARNING:

To C# VS.Net, Java,
To subject "Software engineering"
To <http://programsfactory.univ.kiev.ua> .

Спектр наведених е-ліній технології виробництва програм і ПСпризначений для:

– побудови КПВ і їх розміщення у репозиторію КПВ зі стандартизованим описом паспорту WSDL для їх повторного використання;

- зборки різномовних програм і КПВ у складні ПС;
- взаємодії програм за відповідною моделлю взаємодії програм і систем для різних операційних середовищ, переносу їх інше середовище з функціонуванням даних, що подаються із програм іншого середовища;
- опису онтології доменів в DSL ЖЦ та обчислювальної геометрії засобом Protégé;
- конфігурування готових програм і КПВ у різні варіантні структури ПС
- оцінювання показників якості, вартості та витрат ПС;
- е-програмування мовами C#, Java, Basic, C++ у середовищі VS.Net, Corba та Eclipse;
- навчання основам технології проектування ПС за е-підручником “Програмна інженерія”;
- настанови моделювання доменів засобами інструмента Protégé тощо.

Розділ «Презентації- Presentations» містить прикладну системи підтримки закордонних відряджень в НАНУ, розроблену с.н.с. Куцаченко Л.І. за технологією КП, низку статей на англ. мові по темі виробництва ПС з готового.

Особливе місце в ІТК займає низка нових операцій, пов’язаних з забезпеченням варіантів ПС і СПС шляхом конфігурування КПВ, асетів та змінювання їх по точкам варіантності КПВ в ПС і артефактам СПС з повторним оновленням СПС еквівалентними КПВ.

До складу ІТК включено нові моделі взаємодії через Eclipse (підрозділ «Взаємодія» у сторінки меню веб-сайту за моделями:

- *Visual Studio.Net, Eclipse* на прикладі окремих програм у мові C#, інтерфейсу IDL з перенесенням їх в репозиторій ІТК Eclipse для встановлення зв’язків між ними через конфігураційний файл;
- *Corba, Java, MS.Net* на прикладі програм на МП цих середовищ та встановлення зв’язків між ними через інтерфейс ний посередник в IDL і розміщення їх у репозиторію ІТК;
- *IBM VSphere, МП, Eclipse* розробляється з використанням різних МП, що є у цих середовищах.

І саме головне призначення ІТК – подання підручника по «Програмній інженерії» для електронного навчання студентів теоретичним і прикладним основам цієї дисципліни через наш веб-сайт Інтернету в різних ВНЗ України.

Таким чином, в розробленому веб-сайті ІТК, реалізовані теоретичні аспекти технологій програмної інженерії. Відповідно Google-

статистики до сайту звернулися біля 5000 викладачів та студентів з різних країн

6. Висновки

Сформульовано деякі нові аспекти теоретичного апарату ОКП проектування ПС на основі методу моделювання об’єктної моделі ПрО, механізмів удосконалення теорії КП та переходу от визначених в них базових характеристик і методів об’єктів до програмних компонентів і їх інтерфейсів.

ОКП складається з базових понять об’єктного аналізу ПрО та формального опису базових понять КП (компонент, компонентна модель, модель ПС, модель середовища тощо) та математичних методів збирання КПВ з використанням поширеної зовнішньої та внутрішньої компонентної алгебри.

Розглянуті методи еволюції компонентів та методології компонентного проектування деяких ліній розроблення нових ПС і інформаційних систем.

В комплексну технологію увійшли спрощені лінії виробництва ПС із КПВ, модель взаємодії різномовних КПВ на основі апарата алгебраїчних систем з еквівалентним перетворенням основних типів даних до потрібних форматів середовища виконання та модель варіабельності ПС і СПС.

Крім того, включені оператори еволюції КПВ із алгебри рефакторінгу, реінженерії та реверсної інженерії з операціями їх модифікації. Теоретичні і прикладні нароби є новими, не мають прототипу. Вони подані артефактами і спектром технологій їх підтримки на веб-сайті ІТК <http://sestudy.edu-ua.net> та веб-сайті фабрики програм <http://programsfactory.univ.kiev.ua>, які використовується при навчанні студентів сучасним основам технології зборочного програмування ПС з готових програмних ресурсів (КПВ, reuses, artifacts, servises, assets, general data types тощо).

Основні розробники ОКМ і сайту ІТК зареєстрували його у Державної службі авторського права України [20], друкують статті і доповіді на різних конференціях, зокрема на міжнародній конференції ICTERI-2012, 2013 [21 - 23], а також демонстрували його на виставковому Центрі в 2012, 2013 рр. в День науки України та конференції ICTERI- 2012 и 2013 .

Список використаних джерел

1. *Hrischenko V.* The method of object-component design of software systems // Programming

- problems. – 2007. – №2. – P.113–125. (In Ukrainian)
2. *Lavrishcheva K., Hrischenko V.* The assembly programming. – Kyiv: Naukova dumka, 1991. – 213 p. (In Ukrainian)
3. *Frege G.* The meaning and denotation / Semotika and Informatics. – 1977. – Edit. 8. – P.181–231. (In Russian)
4. *Kolesnik A.* Object analysis of the simulation domain model using the theory of Frege. – TAAPSD–2013. – Proceedings of the Conference. – P.69–74. (In Ukrainian)
5. *Orfali R., Harki D.* Fundamentals of CORBA. – M.: 1999. – 317 p. (In Russian)
6. *Emerich V.* Construction of distributed objects. Techniques and software architectures of interoperable objects in the OMG / CORBA, Microsoft COM and Java RMI. – M.: Мир, 2002. – 510c. (In Russian)
7. *Hrischenko V, Lavrishcheva K.* Methods and tools for component programming // Cybernetics and Systems Analysis, 2003. – №1. – P.39–55. (In Russian)
8. *Hrischenko V* Formal models of component programming / Problems of programming, 2003. – №2. – P.42–57. (In Russian)
9. *Lavrishcheva K.,* Component programming. Theory and Practice // Problems of programming, 2012, №1. – P. 3–14 (In Ukrainian)
10. *Stenyachin A.* The approach to designing distributed software systems with functional and interface objects. – TAAPSD–2013. – Proceedings of the Conference. – P. 173–180. (In Ukrainian)
11. *Lavrishcheva K* Generating programming software systems and families // Problems of programming. – 2009. – № 1. – C. 3 – 16. (In Ukrainian)
12. *Lavrishcheva K* Interaction programs, systems and operating environments // Problems of programming, №3, 2011. – P.13–24. (In Ukrainian)
13. *Lavrishcheva K, Slabospitskaya A., Koval G., Kolesnik A.* Theoretical aspects of managing variability in families of software systems. – Bulletin KNU, Series Physics & Mathematics – 2011. – №1. – P. 151–158. (In Ukrainian)
14. *Kolesnik A.* Approaches to configure reusable components // Problems of programming. – 2011. – № 4. – C. 57 – 66. (In Ukrainian)
15. *Lavrishcheva K.* Instrumental and Technological Development and training techniques of software systems – Kyiv. – Bulletin NAS Ukraine, 2012. – №3. – P.17–26. Available at http://www.nbuv.gov.ua/portal/all/herald/2012_3.pdf (In Ukrainian)
16. *Andon P., Lavrishcheva K.* Development programs factories in the information world // Bulletin of the NAS of Ukraine. – 2010. – № 10. – P. 15–41. Available at <http://www.nbuv.gov.ua/portal/all/herald/2010-10/a3.pdf>. (In Ukrainian)
17. *Lavrishcheva K., Hrischenko V.* The assembly programming. Basics of the software industry. Second-ed. – K.: Naukova Dumka, 2009. – 371p.
18. *Lavrishcheva K.* The paradigm of integration in the program inzhenerii. – Proceedings of the Second International scientific-practical conference. UkProg-2000, 23–26 May, 2000. Kyiv, P.351–360. (In Russian)
19. *Fauler M.* Refactoring: Improving the relevant code. – SPtB.: Symbol-Plus, 2003. – 432 p. (In Russian)
20. *Lavrishcheva K, Zinkovich V., Kolesnik A., Stenyachin A.* Instrumental and a set of technological development and teaching techniques of software systems. – State Intellectual Property Service of Ukraine. – Certificate of registration of copyright. – № 45292, від 27.08.2012. (In Ukrainian)
21. *Lavrishcheva K., Ostrovski A., and Radetskyi I.* Approach to E-Learning Fundamental Aspects of Software Engineering. – Conference ICTERI–12 <http://senldogo0039.springer-sbm.com/ocs/home/ICTERI2012>
22. *Kolesnyk A., Clabospitskaya O.* Tested Approach for Variability Management Enhancing in Software Product Line – Conference ICTERI–12 (In Ukrainian)
23. *Lavrishcheva E., Dzubenko A., Aronov A.* Conception of Programs factory for Representation and E-learning Disciplines of Software Engineering // 9–th International Conf. ICTERI– 2013 “ICT in Education, Research and Industrial Applications; Integration, Harmonization and Knowledge Transfer”, Ukraine, June 17–21, 2013 <http://ceur-ws.org/Vol-1000/>

Надійшла до редколегії 15.07.13

УДК 004.655

Логінов В.¹

PELK: евристичний алгоритм аналізу таблиць ознак

У статті описано евристичний алгоритм розв'язання задачі пошуку породжувальної матриці за відомим набором рядків матриці 0-1. Описано властивості цього алгоритму, які дозволяють розв'язати завдання без обмеження на розмірність матриці та з можливістю доповнення рядків матриці до тих, що вже брали участь у процедурі пошуку.

Ключові слова: евристичний алгоритм, породжувальна матриця, матриця 0-1.

¹Кримський гуманітарний університет, 98635, Україна, АРК, м. Ялта, вул. Севастопольська 2а, e-mail: vladis.logi2011@mail.ru

V. Loginov¹

PELK: heuristic algorithm of analysis of tables of signs

This article describes a heuristic algorithm for the problem of finding a generator matrix solving for the well-known matrix set of lines 0-1. Properties of the algorithm which allows to solve a problem without any restrictions on a matrix dimension and a possibility of matrix lines adding to ones which were previously used in the procedure of search.

Key words: matrix 0-1, heuristic algorithm, originative matrix

¹Crimean University for the Humanities (Yalta), 98635, Ukraine, Crimea, Yalta, Sevastopolskaya, st. 2a, e-mail: vladis.logi2011@mail.ru

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

Постановка задачі [1]:

«Для матриці, яка містить лише елементи 0 та 1, необхідно знайти такі рядки, які неможливо отримати шляхом логічного додавання будь-якої кількості інших рядків цієї матриці».

Математична операція логічного додавання дещо відрізняється від добре нам відомого із курсу елементарної математики додавання чисел «у стовпчик». При логічному додаванні, власне, як і при арифметичному додаванні: $0 + 0 = 0$, $0 + 1 = 1$, та $1 + 0 = 1$. Проте, на відміну від арифметичного додавання, при логічному додаванні $1 + 1 = 1$.

У постановці цієї, начебто, простої задачі, криється один нюанс: якщо для матриці розміром $N \times M$, де N та $M = \text{Const}$, цю задачу можна розв'язати, як то кажуть, «в лоб», методом перебору, то у загальному випадку, коли N та $M \rightarrow \infty$, зробити це перебором неможливо.

Дві евристики, що сформувалися у свідомості автора цілком несподівано, у трамвай, та були ним названі «часткова суперечність за нулем» (ЧСН) та «часткова суперечність за

одиницею» (ЧСО), вирішили цю проблему. Розв'язок, як і зазвичай після його виникнення, виявився дуже простим. На сьогодні, програма, що реалізує алгоритм, потребує півтора десятки операторів мови C++¹. За допомогою цього алгоритму задача вирішується для загального випадку, тобто, при N та $M \rightarrow \infty$.

Опис алгоритму: Нехай кожна аналізована ситуація S_i із набору ситуацій S описується властивим їй набором ознак $W_i \subseteq P$, де P – повний набір ознак S :

$$P = \{P_1, P_2, \dots, P_{n-1}, P_n\} \quad (1)$$

Причому на місці ознаки $P_j \in W_i$ стоїть «1», якщо ця ознака властива для ситуації S_i , та «0», якщо ця ознака у ситуації S_i відсутня:

$$W_i = \{0, 1, 0, \dots, 1, 0\} \quad (2)$$

Таким чином, вся множина S може бути представленою набором описів W :

$$W = \{W_1, W_2, \dots, W_{n-1}, W_n\} \quad (3)$$

У загальному випадку їх можна об'єднати у прямокутну матрицю $W(I, J)$, яка складається, враховуючи (2), із 0 та 1:

де $I = 1, 2, \dots, n$ – номер ситуації S_i із набору S ;
 $J = 1, 2, \dots, m$ – номер ознаки P_j із набору P .

Якщо набір S є великим, та проаналізувати окремо кожну ситуацію $S_i \in S$ неможливо, то необхідно вибрати таку їх підмножину $S_1 \subseteq S$ із набором описів $W_1 \subseteq W$, яка б дозволяла отримувати описи W_i будь-якої ситуації $S_i \in S$ шляхом об'єднання деякої кількості описів $W_{i1} \in W_1$:

$$W_i = W_{i1} \cup W_{i1+1} \cup \dots \cup W_{ik1} \quad (4)$$

Набори $W_i \in W_1$ будемо називати елементарними, а набори W_i , що можуть бути отриманими за допомогою операції об'єднання наборів $W_{i1} \in W_1$ із врахуванням (4) – складеними.

Запропонований алгоритм пошуку підмножини елементарних наборів $W_1 \in W$ базується на послідовній селекції наборів $W_i \in W$.

Нехай набір W_i проходить перевірку на елементарність. З цією метою він послідовно порівнюється з усіма іншими наборами $W_t \in W$, де $t = 1, 2, \dots, n; t \neq i$.

Введемо наступне означення:

1. Часткова суперечність за нулем (ЧСН) або часткова суперечність за одиницею (ЧСО) виникають, коли при співставленні значень ознак набору, що порівнюється, $W_i \in W$ та чергового набору $W_t \in W$, $t \neq i$, значення ознаки PJ , у наборах W_i і W_t не співпадають;

2. Повна суперечність за нулем (ПСН) або повна суперечність за одиницею (ПСО) виникають, якщо ЧСН або ЧСО відповідно визначаються на при кожному кроці порівняння значень ознаки PJ набору W_i , що перевіряються, та всіх інших наборів $W_t \in W$, $t = 1, 2, \dots, n; t \neq i$.

Виникнення ПСН або ПСО є достатньою умовою для того, щоби стверджувати, що набір, що перевіряється, W_i , є елементарним, оскільки значення хоча б однієї із його ознак, що входять до набору W_i , у даному випадку його ознаки PJ , не може бути отримано в результаті об'єднання будь-яких наборів $W_t \in W$, $t \neq i$.

У випадку, коли на даному кроці порівняння значень PJ у наборах W_i та W_t ЧСО відсутня, необхідно перевірити відсутність ЧСН між набором W_i , що перевіряється, та поточним набором $W_t \in W$ для всіх ознак $PJ \in W_i$, що мають значення 0. Якщо вони не мають ЧСН, то можна стверджувати, що W_t є одним із наборів, що при поєднанні дадуть набір W_i .

Виконання цієї умови для кожної ознаки $PJ \in W_i$, що дорівнює 1, є достатнім, щоби стверджувати, що набір W_i , що перевіряється, є складеним із множини наборів W . Якщо для деякої ознаки $PJ \in W_i$, що дорівнює 1, виявиться,

що виконати цю умову неможливо, то це є достатнім, щоби стверджувати, що набір $W_t \in W$ є елементарним.

При порівнянні наборів W_i і W_t може виявитися, що ЧСО та ЧСН є відсутніми для кожної ознаки $PJ \in W_i$. Тоді можна стверджувати, що ці набори є еквівалентними. Результати проведеного аналізу дають можливість створити ефективний алгоритм для пошуку підмножини елементарних наборів

$W_1 \subseteq W$.

Алгоритм складатиметься із наступних кроків [2].

Крок 0. [Початкова установка для I] $\leftarrow I_1$, цикл по I , що складається із кроків $M_2 - M_{16}$, це найбільш зовнішній із вкладених циклів.

Крок 1. [Установка початкових значень] (Цикл по J). $J_1, K_1 \leftarrow 0, K_0 \leftarrow 0$.

Крок 2. [Перевірка значення ознак W_i]. Якщо $J > \text{LANG}$, перейти до кроку 5, якщо $\text{ISX}(I, J) = 0$, то перейти до кроку 4.

Крок 3. [Підрахунок кількості і визначення адрес одиниць у наборі W_i] $K_1 \leftarrow K_1 + 1$, $\text{KOL}_1(K_1) \leftarrow J$, $J \leftarrow J + 1$, перейти до кроку 2.

Крок 4. [Підрахунок кількості і визначення адрес одиниць у наборі W_i] $K_0 \leftarrow K_0 + 1$, $\text{KOL}_0(K_0) \leftarrow J$, $J \leftarrow J + 1$, перейти до кроку 2.

Крок 5. [Початкова установка для K] (Цикл по K) $\leftarrow K_1$.

Крок 6. [Визначення позиції поточної одиниці в наборі W_i] $L_1 \leftarrow \text{KOL}_1(K)$.

Крок 7. [Начальна установка для L] (Цикл по L). $L \leftarrow 1$.

Крок 8. [Виключення порівняння W_i з W_i]. Якщо $L = I$, то перейти до кроку 14. Якщо $L \neq I$, то перейти до кроку 9.

Крок 9. [Перевірка співпадіння значень ознак]. Якщо $\text{ISX}(L, L_1) = 1$, то перейти до кроку 10. Якщо $\text{ISX}(L, L_1) \neq 1$, то перейти до кроку 14.

Крок 10. [Начальна установка для M] (Цикл по M). $\leftarrow M_1$.

Крок 11. [Визначення позиції поточного нуля в наборі W_i]. $M \leftarrow \text{KOL}_0(M)$.

Крок 12. [Перевірка співпадіння значень ознак]. Якщо $\text{ISX}(L, M_1) = 0$, то перейти до кроку 13. Якщо ні, то перейти до кроку 14.

Крок 13. [Перехід до наступного кроку порівняння]. $M \leftarrow M + 1$, якщо $M > K_0$,

то перейти до кроку 14, а якщо ні, то перейти до кроку 11.

Крок 14. [Перевірка вичерпання масиву наборів W]. $L \leftarrow L+1$, якщо $L > KOL$, то перейти до кроку 17, якщо ні, то перейти до кроку 8.

Крок 15. [Перехід до перевірки можливості отримання наступної ознаки набору W_i]. $K = K + 1$, якщо $K > K1$, то перейти до кроку 16, якщо ні, то перейти до кроку 6.

Крок 16. [Перехід до перевірки можливості отримання наступної ознаки набору W_i]. $I \leftarrow I+1$, якщо $I > KOL$, перейти до кроку 19, якщо ні, то перейти до кроку 1.

Крок 17. [Перезапис коду W_i у масив елементарних кодів]

Крок 18. [Перехід до кроку 16]. Продовження пошуку.

Крок 19. [Зупинка алгоритму].

Де $ISX(I,J)$ – масив наборів $W_i \in W$;

KOL – кількість наборів $W_i \in W$;

$LANG$ – кількість ознак у наборі W_i ;

$K1$ – кількість одиниць у наборі W_i ;

$KOL1(I)$ – масив адрес одиниць в наборі W_i ;

$K0$ – кількість полів у наборі W_i ;

$KOL0(I)$ – масив адрес полів у наборі W_i .

Висока швидкість роботи даного алгоритму визначається:

- ефективним зменшенням області пошуку у порівнянні із повним перебором можливих варіантів поєднання наборів по 2, 3, 4, ..., і порівнянням результатів такого поєднання із вихідним набором ознак до отримання повного співпадіння із будь-яким набором W_i , що представлений у W , але не приймає участі у поєднанні;
- використанням при застосуванні алгоритму лише операцій порівняння та пересилки, які виконуються швидше, ніж інші операції із складу команд ПК.

При розгляді матриці W_{iJ} , яку складають значення ознак P_{iJ} множини ситуацій W

$$W_{iJ} = \begin{vmatrix} P_{11} & P_{12} & \dots & P_{1l} & \dots & P_{1m} \\ P_{21} & P_{22} & \dots & P_{2l} & \dots & P_{2m} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ P_{k1} & P_{k2} & \dots & P_{kl} & \dots & P_{km} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ P_{n1} & P_{n2} & \dots & P_{nl} & \dots & P_{nm} \end{vmatrix} \quad (5)$$

можливо отримати наступні наслідки проведеного аналізу:

Наслідок 1. Якщо набір W_i є елементарним для будь-якої частини стовпців матриці W_{iJ} (наприклад, для стовпців P_{iJ} , $J=1,2,\dots, L$), то він є елементарним для всієї матриці W_{iJ} .

ПСН та ПСО є прикладами істинності даного наслідку коли кількість стовпців матриці W_{iJ} дорівнює 1.

Наслідок 2. Якщо набір W_i є складеним на будь-якій частині матриці W_{iJ} , наприклад, для рядків $i = 1, 2, \dots, k$, то він є складеним і для всієї матриці W_{iJ} .

Звідси випливає, що великі за кількістю стовпців та рядків матриці можуть бути проаналізовані по частинам. Це знімає обмеження на об'єм масивів в програмі, яка використовує даний алгоритм.

Окрім того, до отриманої підмножини елементарних наборів $W1$, що характеризують у певному сенсі всю множину W , можна додавати будь-яку кількість наборів, що описують нові ситуації, для спільного аналізу.

При цьому на підставі наслідку 2, можна стверджувати, що набори, які входять в це поєднання, і які є елементарними у ньому, також є елементарними і для кожної підмножини наборів у цьому поєднанні. Зворотне твердження в загальному випадку є невірним.

Розглянемо приклад. Нехай матриця наборів ознак W_{iJ} має вигляд:

$$W_{iJ} = \begin{vmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{vmatrix}$$

Виділимо перший набір $W1=\{0,0,1,0,1\}$, і визначимо місця нулів та одиниць у ньому, записавши їхні адреси у масиви $KOL1=\{3,5\}$ та $KOL0=\{1,2,4\}$. Виберемо перший адрес одиниці із масиву $KOL1$ та, порівнюючи значення ознак $P3$ в усіх інших наборах, визначимо, чи існують інші набори, для яких $P3 = 1$. У нашому випадку це буде набір $W2$. Перевіримо відсутність ЧСН, тобто, чи не містяться у наборі $W2$ одиниці на тих місцях, де в наборі $W1$ стоять нулі. ЧСН жодного разу не зустрілося у процесі порівняння. Це дає підставу включити набір $W2$ в поєднання для отримання набору $W1$. У наборі $W1$ наступна

одиниця має адресу 5. Порівняємо значення ознак P5 у всіх інших наборах. Набір W3 має ознаку P5 = 1, але він ЧСН за ознаками P2 і P4, що є підставою не включати його в поєднання для отримання набору W1. Зазначимо також, що достатньою підставою для цього рішення є наявність одного (першого) ЧСН.

Наступний набір, що має ознаку P5 = 1, є набір W4. Перевірка показує, що одночасно він не дає ЧСН за ознаками P1, P2 та P4. Це достатня умова для того, щоби не включати його в шуканий набір. Інших ознак, окрім P3 та P5, що мають значення 1 в наборі W1 складений, адже він може бути отриманим поєднанням наборів ознак W2 та W4:

$$W1 = W2 \cup W4$$

Подальший аналіз показує, що окрім набору W1, складеним буде і набір W3. Інші набори є елементарними. В $W1 \subseteq W$ входять набори W2, W4 та W5:

$$W1iJ = \begin{vmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{vmatrix}$$

Програма PELK, що реалізує даний алгоритм, спершу була написана на мові ФОРТРАН для машин типу: ЕС ЕОМ, СМ-1420, та була переписана на СІ++, для персональних комп'ютерів типу ІВМ. Роки використання даної програми показали її високу надійність та ефективність.

Перед нами не стоїть мета довести відповідність рішення всім вимогам, що висуваються до математичних алгоритмів. Цікаво подивитись, в яких задачах алгоритм PELK може допомогти у пошуку відповіді, окрім, звичайно, тої, для розв'язку якої він був створений. Тут необхідно враховувати можливість переформулювання задачі, тобто, застосування для її опису мови не тої галузі знання, для якої вона була вперше сформульована.

Якщо припустити, що наявність деякої ознаки чи якості у описі предмету чи суб'єкту діяльності, ситуації, вимоги до особистості, або до професійної діяльності може бути записаною як 1 у таблиці алгоритмів ознак, а її відсутність – як 0,

то номенклатура можливостей PELK різко зростає.

З його допомогою:

- можна вирішувати чисто педагогічну задачу вибору із множини можливих завдань такого завдання, в якому присутній новий елемент або їх незвичайне поєднання. Або необхідний мінімальний комплекс завдань, що поширюється на все поле викладеного матеріалу;
- можна вибирати із множини психологічних тестів, які спрямовані на дослідження більш ніж одного особистісного фактору, мінімальну їх кількість, достатню для аналізу необхідних якостей особи;
- із множини кандидатів для виконання деякої задачі можна обирати необхідну мінімальну їх кількість, яка б забезпечила створення групи співробітників із різноманітними особистими якостями та різним рівнем підготовки, які могли б вирішити всі задачі, що можуть виникнути перед цією групою. Наприклад, проводити підбір кадрів для нової фірми, спираючись на опис її діяльності та особистісних і професійних якостей, які ця діяльність вимагає від майбутніх співробітників;
- можна навіть визначити необхідний мінімальний набір номіналів монет, що дозволяє розраховуватись будь-якою сумою при мінімальній кількості монет у грошовому оберті та ін.

Найбільш широке застосування цей алгоритм може знайти у системах адаптивної освіти (ADOS) для вибору педагогічних засобів, які найліпше відповідають матеріалу, що вивчається.

Висновки

1. У статті описаний евристичний алгоритм пошуку матриці, що породжує, по відомому набору рядків матриці 0-1.
2. При реалізації алгоритму використовуються лише операції порівняння і пересилки, що різко підвищує швидкість його роботи.
3. Запропонований алгоритм може бути використаний при вирішенні оптимізаційних завдань незалежно від їх наочної області.

Список використаних джерел

1. "Algorithm of the tables characteristics analysis" 9 p., UkrNIINTI, deposited 17.12.85, № 2744-Uk85.

2. "Digital and analog converter and a converter setting method", 7 p., Copyright certificate №1324114 from 15.03.1987.

Надійшла до редколегії 26.06.2013

УДК 004.021: 004.312.4: 004.421.6: 004.414.2

Лук'янова О.О., к.ф.-м.н, доц.

**Про бісимуляційну еквівалентність
детальної моделі Петрі та її CN-моделі
досліджуваної паралельної розподіленої
системи**

Вивчається поведінкова еквівалентність компонентної мережі Петрі (CN-моделі) моделі детальної мережі Петрі N , що відображує досконале функціонування досліджуваної паралельної розподіленої системи. Моделі N і CN подаються з точки зору реляційних моделей, і через визначення гомоморфізму графів досяжних розміток досліджуваних моделей Петрі встановлюється бісимуляція досліджуваних N і CN моделей.

Ключові слова: мережа Петрі, компонентна мережа Петрі, граф досяжних розміток, бісимуляційна еквівалентність.

Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Глушкова 4д,
e-mail: lukyanovaea@mail.ru

Статтю представив д. ф.-м. н., проф. Буй Д.Б.

Lukyanova E.A., PhD

**On Bisimulation Equivalence of Detailed
Petri Model and Its CN-Model of Investigated
Parallel Distributed System**

The behavioral equivalence of component Petri net (CN-model) to detailed Petri net model N has been studied, that reflects a thorough operation of investigated parallel distributed system. Models N and CN has been given in terms of relational models, and through the detection of homomorphism of graphs of reachable markings in investigated Petri nets, bisimulation of investigated N and CN models has been determined.

Key words: Petri Net, componental Petri Net, graphs of reachable markings, bisimulation equivalence.

Taras Shevchenko National University of Kyiv,
03680, Kyiv, Glushkova st., 4d,
e-mail: lukyanovaea@mail.ru

Вступ

Математичне моделювання є загальноприйнятим засобом дослідження складних систем. В якості формальної моделі для отримання найбільш повної інформації про систему продуктивно використовується формалізм мереж Петрі. Такі моделі здатні відображати всі аспекти, що цікаві для аналізу модельованої системи, і зберігати всі властивості модельованої системи, що зрештою дає змогу якісно проводити процес верифікації. Одним з найбільш важливих підходів до верифікації мереж Петрі є аналіз простору станів мережі і встановлення можливості еквівалентних перетворень роботи мережі. Наприклад, таких як симуляційна і бісимуляційна еквівалентності.

Відносини симуляції або відносини подібності спеціального виду [1, 2], що виникають у задачах верифікації, є бінарними відношеннями, які задані на множині моделей обчислення, що дає

можливість порівнювати структуру дерева обчислень однієї моделі зі структурою дерева обчислень іншої моделі.

Для порівняння поведінки паралельних розподілених систем було введено [3, 4, 5] поняття бісимуляційної еквівалентності, яке використовується при встановленні подібності мереж Петрі в результаті їх редукції. Бісимуляційна редукція мережі Петрі дає змогу отримувати мережі менших розмірів при збереженні поведінкових властивостей вихідної первісної мережі. А побудова адекватної моделі значно менших розмірів вихідної моделі дає можливість вирішувати задачу експоненційної складності більшості алгоритмів аналізу мереж Петрі великих розмірів. Тому пошук еквівалентних з точки зору бісимуляційних структур мереж Петрі є актуальним завданням, рішення якого дає змогу істотно полегшити як аналіз моделі, так і аналіз досліджуваної системи.

Компонентне моделювання [6, 7] можна розглядати як метод редукції [9] детальних моделей Петрі паралельних розподілених систем.

Метою даної роботи є обґрунтування бісимуляційної еквівалентності компонентної мережі Петрі (CN-мережі) вихідної детальної моделі Петрі N досліджуваної паралельної розподіленої системи.

1. Постановка задачі

Бісимуляційна еквівалентність мереж Петрі традиційно вивчається на їх графах досяжних розміток. Тому розглядаючи в якості моделі паралельної розподіленої системи детальну мережу Петрі N , що досконально моделює функціонування досліджуваної системи, перейдемо до графа досяжних розміток цієї мережі. Такий граф повністю характеризує динаміку функціонування побудованої детальної мережі Петрі і відображає всі можливі зміни розміток мережі N , що відбуваються в результаті спрацювання її переходів [8].

Позначимо орієнтований граф досяжних в мережі N розміток через G_1 . Для моделі системи у вигляді CN-мережі її граф досяжних розміток позначимо через G_2 . Згідно з правилами побудови CN-мережі [6, 7] складові компоненти в CN-мережі є або місцем, або переходом, а для детальної моделі N вони є ділянкою цієї мережі. Зрозуміло, що при переході до графів досяжних розміток, у графі G_2 досяжних розміток CN-мережі деякі вершини будуть акумулювати цілі ділянки графа G_1 досяжних розміток детальної моделі N досліджуваної системи. Це означає, що для встановлення бісимуляційної еквівалентності вищевказаних мереж Петрі необхідно розглянути й питання, що стосуються станів ділянок графа досяжних розміток мережі N , які відображають динаміку функціонування складових компонент.

2. Приклад графів досяжних розміток для детальної та компонентної мереж Петрі

На рис. 1, а) показано скорочений фрагмент моделі залізничного вузла з [7], що становить зовсім невелику компонентну мережу. Ця компонентна мережа містить дві однакові складові компоненти – компоненти-переходи T_1 і T_2 , внутрішня структура яких показана на рис.1, б).

Зображена на рис. 1, а) мережа отримана з відповідної детальної мережі Петрі, що показана

на рис. 2, в результаті оформлення деяких її однакових ділянок у компоненти-переходи згідно з визначеннями складових компонент даних в [10].

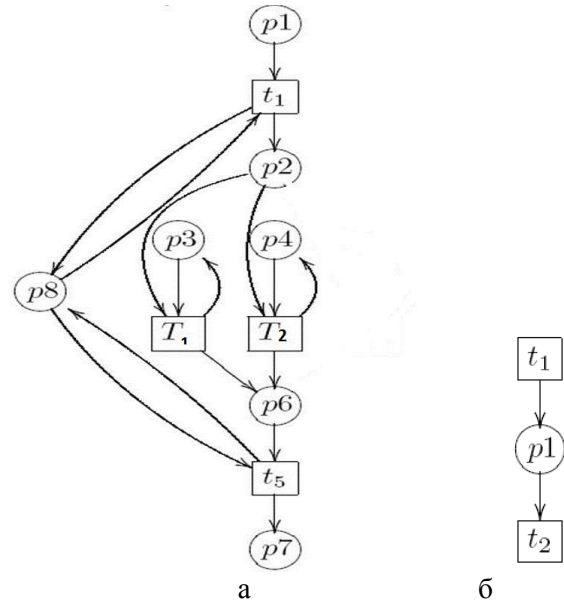


Рис. 1. А) Компонентна мережа, що містить однакові компоненти-переходи T_1 і T_2 ;
б) внутрішня структура однакових компонент-переходів T_1 і T_2

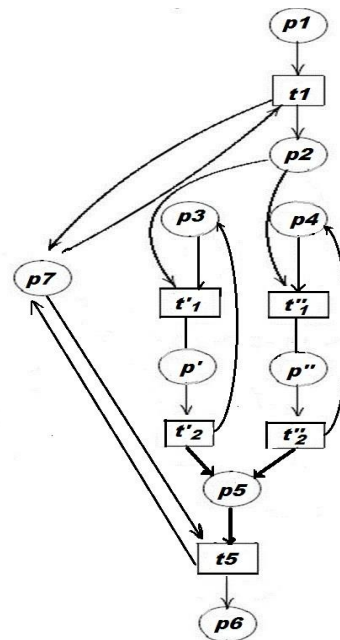


Рис. 2. Мережа Петрі, що є детальною моделлю для компонентної мережі Петрі з рис. 1, а)

На рис. 3 та рис. 4 показані графи досяжних розміток G_1 і G_2 відповідно для детальної

мережі з рис. 2 і компонентної мережі з рис. 1, а).

Ділянки G'_1 і G'_2 графа досяжних розміток G_1 детальної моделі Петрі, які відмічені на рис. 3, відображають динаміку функціонування відповідних однакових складових компонент – компонент-переходів T_1 і T_2 , що у графі досяжних розміток компонентної мережі інкапсулюються у відповідні вершини a'_3 і a'_4 графа G_2 .

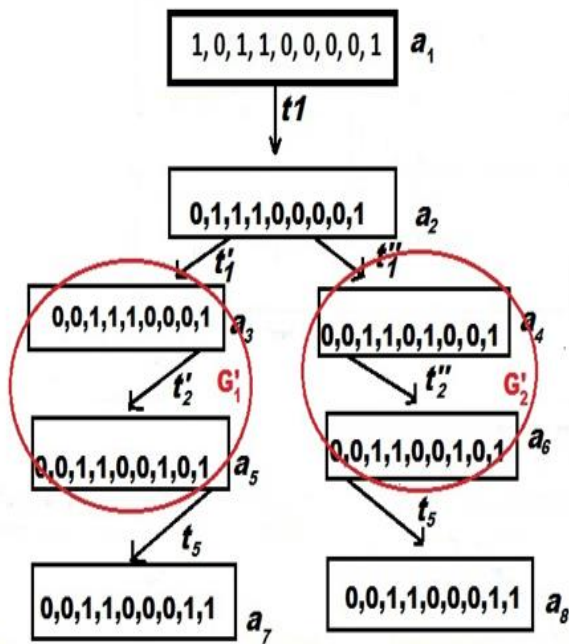


Рис. 3. Граф розміток G_1 детальної моделі Петрі з рис. 2

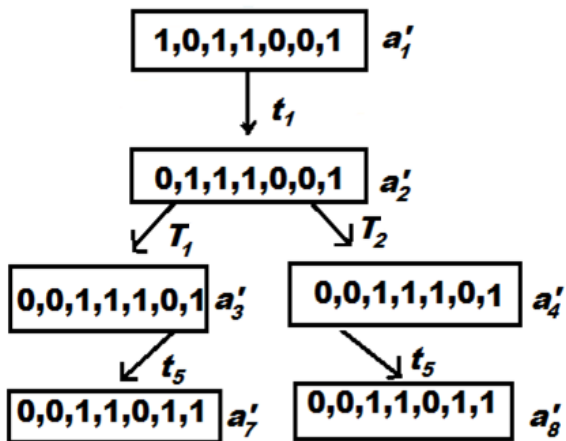


Рис. 4. Граф розміток G_2 компонентної мережі Петрі з рис. 1, а)

3. Графи досяжних розміток детальної і компонентної мережі Петрі як алгебраїчні системи

Розглянемо введені в [9] відношення компоненти χ тепер на множині вершин графа досяжних розміток детальної моделі Петрі N . Позначимо його χ_1 . Згідно з [9] відношення компоненти χ_1 на множині вершин графа G_1 досяжних розміток детальної моделі Петрі має задовольняти таким умовам:

- 1) для кожної вершини $a \in G_1$ виконується $a\chi_1 a$;
- 2) для двох вершин a і $b \in G_1$ виконується $a\chi_1 b$, якщо a і $b \in G'_i$ є вершинами однієї ділянки графа досяжних розміток мережі N , що відображає динаміку функціонування відповідної складової компоненти.

Розглянуте на множині вершин графа досяжних розміток детальної моделі Петрі N відношення χ_1 є рефлексивним, симетричним, транзитивним бінарним відношенням, яке розбиває множину вершин графа досяжних розміток детальної моделі Петрі на класи еквівалентності, що не перетинаються. У кожен такий клас потрапляють вершини, для яких виконуються умови 1) – 2). Якщо вершина графа G_1 не є вершиною ні однієї з ділянок графа досяжних розміток мережі N , що відображає динаміку функціонування складової компоненти, то ця вершина сама представляє клас еквівалентності – одиничний клас.

З кожним орієнтованим графом пов'язане несиметричне бінарне відношення – відношення суміжності, що задає впорядковані пари вершин графа, для яких у графі є дуга, що з'єднує їх. Тоді графи розміток G_1 і G_2 можна розглядати з алгебраїчної точки зору, як моделі (реляційні системи) з сигнатурою, що складається з двох бінарних відношень: відношення ρ – відношення суміжності і відношення χ – відношення компоненти. З цього випливає, що граф розміток $G_1 = (V_1, \rho_1, \chi_1)$ і граф розміток $G_2 = (V_2, \rho_2, \chi_2)$ з відповідними множинами вершин V_1 і V_2 , є моделі з бінарними відношеннями ρ_1 , ρ_2 , і χ_1 , χ_2 , які отримані за такою ж схемою з відповідних базисних множин V_1 і V_2 . Ці моделі є однотипними алгебраїчними системами [11].

Зазначимо, що в результаті вищеописаної побудови графа G_2 для відношення χ_2 умова 2) фактично вироджується в умову 1).

4. Гомоморфізм графів досяжних розміток детальної і компонентної мереж Петрі

Відображенням однієї алгебраїчної системи в іншу алгебраїчну систему називається відображення базисної множини першої системи в базисну множину другої системи [11].

Розглянемо відображення $h: V_1 \rightarrow V_2$ множини вершин графа $G_1 = (V_1, \rho_1, \chi_1)$ в множину вершин графа $G_2 = (V_2, \rho_2, \chi_2)$. При цьому відображенні кожен клас еквівалентності, побудований на множині вершин графа G_1 по відношенню χ_1 , а, значить, кожна вершина відповідної ділянки графа досяжних розміток мережі N , що відбиває динаміку функціонування відповідної складової компоненти, відображається у відповідну одну вершину графа G_2 зі збереженням умови суміжності для образів вершин графа G_1 .

Дійсно, для однойменних відносин χ_1 і χ_2 та ρ_1 і ρ_2 й для будь-якої пари елементів a_1, a_2 із V_1 при відображенні $h: V_1 \rightarrow V_2$ мають місце такі імплікації:

$$h(\chi_1(a_1, a_2)) \rightarrow \chi_2(h(a_1), h(a_2)) \quad (1),$$

$$h(\rho_1(a_1, a_2)) \rightarrow \rho_2(h(a_1), h(a_2)) \quad (2).$$

Розглянемо умову (1). Якщо обидва або хоча б один з елементів a_1, a_2 не є елементами G'_i – однієї з ділянок графа досяжних розміток мережі N , що відображає динаміку функціонування складовою компоненти, тоді відношення $a_1 \chi_1 a_2$ хибне, а значить імплікація (1) тривіальна. Якщо обидва елементи $a_1, a_2 \in G'_i$, то відношення $a_1 \chi_1 a_2$ є істиною і $h(a_1) = h(a_2)$, значить згідно з умовами 1) – 2) відношення $\chi_2(h(a_1), h(a_2))$ є істиною і є істиною імплікація (1).

Розглянемо умову (2). Якщо відношення $a_1 \rho_1 a_2$ хибне, тобто вершини a_1, a_2 у графі G_1 не є суміжними, то імплікація (2) тривіальна.

Якщо відношення $a_1 \rho_1 a_2$ є істиною, то мають місце такі можливості:

а) вершини a_1 і a_2 графа G_1 не належать ніякій ділянці G'_i , а значить, є одноелементними класами еквівалентності у відношенні χ_1 , тоді відношення $h(a_1) \rho_2 h(a_2)$ є істиною, отже, імплікація (2) є істиною;

б) одна з вершин a_1 або a_2 належить деякій ділянці G'_i , тобто є елементом деякого класу еквівалентності за відношенням χ_1 , а значить, при відображенні h є елементом повного прообразу свого образу, тоді відношення $h(a_1) \rho_2 h(a_2)$ є істиною і імплікація (2) є істиною;

в) обидві вершини a_1 і a_2 належать одній ділянці G'_i , тобто є елементами одного класу еквівалентності за відношенням χ_1 , тоді при відображенні h образи цих вершин збігаються ($h(a_1) = h(a_2)$) і є однією і тією вершиною графа G_2 . Ця вершина згідно з умовами 1) – 2) сама з собою знаходиться у відношенні χ_2 : $h(a_1) \chi_2 h(a_2)$. Отже, у випадку в) отримаємо наступний запис:

$$(h(\rho_1(a_1, a_2)) \rightarrow h(\chi_1(a_1, a_2))) \wedge \\ \wedge (h(\chi_1(a_1, a_2)) \rightarrow \chi_2(h(a_1), h(a_2))),$$

який з урахуванням того, що вершини a_1 і a_2 суміжні в G_1 і є елементами однієї ділянки G'_i графа досяжних розміток мережі N , приймає значення істина. Отримано, що в результаті введення на множині вершин графа досяжних розміток детальної моделі Петрі N відношення еквівалентності χ_1 при відображенні h у випадку в) у графі G_2 не з'являться петлі при вершині $h(a_1)$ і імплікація (2) приймає вигляд (1) і є істиною.

Більше того, для однойменних відношень ρ_1 і ρ_2 та χ_1 і χ_2 відповідно з $G_1 = (V_1, \rho_1, \chi_1)$ і $G_2 = (V_2, \rho_2, \chi_2)$ мають місце наслідки (3) та (4):

$$\begin{aligned} & \rho_2(h(a_1), h(a_2)) \rightarrow \\ & \rightarrow \exists_{V_1} b_1, b_2 ((h(a_k) = h(b_k), k = 1, 2) \wedge \\ & \wedge (\rho_1(b_1, b_2))), \end{aligned} \quad (3),$$

де a_1, a_2 елементи множини V_1 : або обидва елементи $a_1, a_2 \notin G'_i$, або хоча б один з елементів a_1 або a_2 не є елементами G'_i – однієї з ділянок графа досяжних розміток мережі;

$$\begin{aligned} & \chi_2(h(a_1), h(a_2)) \rightarrow \\ & \rightarrow \exists_{V_1} b_1, b_2 ((h(a_k) = h(b_k), k = 1, 2) \wedge \\ & \wedge (\rho_1(b_1, b_2))), \end{aligned} \quad (4),$$

де a_1, a_2 елементи множини V_1 : обидва елементи $a_1, a_2 \in G'_i$ – одній з ділянок графа досяжних розміток мережі N , і $h(a_1) = h(a_2)$, $b_1 \chi_1 b_2$.

Доведено теорему 1.

Теорема 1. Нехай на множині вершин графа G_1 досяжних розміток детальної моделі Петрі N задано відношення компоненти χ_1 , тоді відображення $h: V_1 \rightarrow V_2$ множини вершин графа $G_1 = (V_1, \rho_1, \chi_1)$ в множину вершин графа $G_2 = (V_2, \rho_2, \chi_2)$ є гомоморфізмом.

5. Бісимуляція моделей Петрі

Для порівняння поведінки розглянутих мереж по отриманому гомоморфізму зіставимо деяке відношення – бісимуляцію, яка встановлює рівень еквівалентності у функціонуванні мереж N і CN . Розглядати бісимуляцію будемо як відношення, що задане на множині вершин об'єднання графів досяжних розміток G_1 і G_2 мереж N і CN .

Введемо попередні означення та позначення. На множині вершин графа досяжних розміток G деякої мережі Петрі MP визначимо відношення переходу як трійку (a, μ, a') або знакосполучення $a \xrightarrow{\mu} a'$, де a, a' – вершини G , μ – ребро з a в a' .

Позначимо знакосполученнями:

1) $a_1 \xrightarrow{\tau^*} a_2$, коли або $a_1 = a_2$, або в G є шлях з a_1 в a_2 з ребрами, поміченими τ ;

2) $a_1 \xrightarrow{\mu^*} a_2$, коли або $\mu = \tau$ і $a_1 = a_2$, або

$a_1 \xrightarrow{\tau^*} a'_1 \xrightarrow{\mu} a'_2 \xrightarrow{\tau^*} a_2$, де a'_1, a'_2 деякі вершини G .

Розглянемо мережі N і CN , їх графи досяжних розміток $G_1 = (V_1, \rho_1, \chi_1)$ і $G_2 = (V_2, \rho_2, \chi_2)$ і бісимуляцію $R \subseteq V_1 \times V_2$. При цьому для бінарного відношення $R \subseteq V_1 \times V_2$ на множині вершин $V_1 \cup V_2$ об'єднання графів досяжних розміток G_1 і G_2 мереж N і CN повинні виконуватися умови:

1) для початкової вершини $a_{01} \in G_1$ (початкової розмітки мережі N) існує початкова вершина $a_{02} \in G_2$ (початкова розмітка мережі CN): $(a_{01}, a_{02}) \in R$;

2) для початкової вершини $a_{02} \in G_2$ існує початкова вершина $a_{01} \in G_1$: $(a_{02}, a_{01}) \in R$;

3) для кожної пари вершин $(a_1, a_2) \in R$ ($a_1 \in V_1, a_2 \in V_2$) і кожної вершини $a'_1 \in V_1$ такої, що $a_1 \xrightarrow{\mu} a'_1$ для деякої дії μ існує вершина $a'_2 \in V_2$: $(a'_1, a'_2) \in R$ і $a_2 \xrightarrow{\mu^*} a'_2$;

4) для кожної пари вершин $(a_1, a_2) \in R$ ($a_1 \in V_1, a_2 \in V_2$) і кожної вершини $a'_2 \in V_2$ такої, що $a_2 \xrightarrow{\mu} a'_2$ для деякої дії μ існує вершина $a'_1 \in V_1$: $(a'_1, a'_2) \in R$ і $a_1 \xrightarrow{\mu^*} a'_1$.

Теорема 2. Нехай компонентна мережа Петрі CN виходить з детальної мережі Петрі N за допомогою редукції, що складається з виділення складових компонент, тоді мережі N і CN бісимулярні.

Доведення теореми ґрунтується на тому, що відношення $R \subseteq V_1 \times V_2$ розбиває множину вершин $V_1 \cup V_2$ на класи еквівалентності за відношенням R наступним чином: кожен клас еквівалентності за відношенням R містить вершини з V_1 і V_2 , такі, що вершини з V_1 належать одному класу еквівалентності за відношенням χ_1 , а вершини з V_2 є

гомоморфними образами цих класів еквівалентності при відображенні h .

Наслідок. Якщо ділянки графа досяжних розміток мережі N , що відображають динаміку функціонування складових компонент, не містять тупикових розміток, то за властивостями, які можуть бути встановленими на графі досяжних розміток мережі CN , можна встановлювати ті ж самі властивості і всієї детальної мережі N .

Зауваження. В умовах висновку достатньо перевіряти відсутність тупиків тільки для однієї ділянки, що відображає динаміку функціонування одного представника з групи однотипних складових компонент з найменшим числом однакових паралельних процесів.

Висновок

Ефективне для встановлення та аналізу структурних властивостей [7, 12], зручне для вивчення мовних характеристик [13, 14] розширення мереж Петрі у вигляді компонентної мережі Петрі у цій статті вивчалось з точки зору поведінкових еквівалентностей щодо детальної моделі Петрі досліджуваної паралельної розподіленої системи. З'ясовано, що редукція детальної моделі Петрі, яка полягає у виділенні складових компонент [10], є бісимуляційною редукцією. Запропоноване поняття бісимуляції для детальної моделі Петрі N і компонентної моделі CN визначалося через встановлення гомоморфізму графів досяжних розміток досліджуваних мереж Петрі. Розглянута бісимуляційна редукція дає змогу отримувати нові адекватні моделі досліджуваних систем, які мають значно менші розміри при збереженні поведінкових властивостей (у розумінні бісимуляції) вихідної детальної моделі досліджуваної системи.

Список використаних джерел

1. Henzinger M. R. Computing simulations on finite and infinite graphs / M. R. Henzinger, T. A. Henzinger, P. W. Kopke P. W. // 37th IEEE Symp. On Foundations of Comp. Sci. – IEEE Computer Society Press, 1996. – P. 453–462.
2. Groote J.F. Vaandrager F.W. An efficient algorithm for branching bisimulation and stuttering equivalence / J.F. Groote // Automata, Languages and Programming. – 1990. – P. 626–638.
3. Milner R. An algebraic definition of simulation between programs / R. Milner // Tech.rep. (Report No.CS205) – Stanford, CA, USA: 1971. – P. 481–489.
4. Milner R. A Calculus of Communicating Systems / R. Milner // LNCS 92, 1980. – 171 p.
5. Park D. Concurrency and automata on infinite sequences, LNCS 104, 1981. – P. 168–183.
6. Lukyanova E. A. On component modeling of systems with concurrency / E. A. Lukyanova // Naukovi Zapysky of NaUKMA. Computer Science. – 2012. – T. 138. – P. 47–52. (in Ukrainian).
7. Lukyanova E. A. The study of the structural elements of the single-type CN -network during the component modeling and analysis of complex systems with concurrency / E. A. Lukyanova, A. V. Dereza // Cybernetics and Systems Analysis. – 2012. – № 6. – P. 20–29. (in Russian).
8. Kotov V. E. Petri nets / V. E. Kotov – Moscow: Nauka, 1984. – 160 p. (in Russian).
9. Lukyanova E. A. On homomorphism of component Petri net / E. A. Lukyanova // Cybernetics and Systems Analysis. – 2014. – № 1. – P. 16–25. (in Russian).
10. Lukyanova E. A. The structural elements of a Petri net component / E. A. Lukyanova // Problemy programuvannya. – 2012. – № 2-3. – P. 25–32. (in Russian).
11. Maltsev A.I. Algebraic systems / A.I. Maltsev – Moscow: Nauka, 1970. – 392 p. (in Russian).
12. Lukyanova E. A. On component analysis of parallel distributed systems / E. A. Lukyanova // TVIM. – 2011. – № 2. – P. 71–81. (in Russian).
13. Lukyanova E. A. On the relationship between the language of CN -Model with component junctions, and the language of detailed Petri Model of parallel distributed system/ E. A. Lukyanova // Visnyk of Kyiv Univ. im. Tarasa Shevchenka. – 2012. – № 4. – P. 145–150. (in Ukrainian).
14. E. Lukyanova Component modeling: on connections of detailed Petri model and component model of parallel distributed system / E. A. Lukyanova // ITHEA. – 2013. – P. 15–22.

Надійшла до редколегії 01. 06. 2013

УДК 004.42:510.69

Нікітченко М.С., д. ф.-м. н., проф.,
Шкільняк С.С., д. ф.-м. н.

**Композиційно-номінативні логіки
із спеціальними предикатами наявності
значення для змінних**

Досліджено чисті першопорядкові композиційно-номінативні логіки, розширені спеціальними предикатами-індикаторами наявності значення для змінних. Розглянуто семантичні властивості таких логік, пов'язані з цими предикатами. Для різних відношень логічного наслідку в таких логіках часткових однозначних, тотальних неоднозначних та часткових неоднозначних предикатів побудовано числення секвенційного типу, доведено коректність і повноту цих числень.

Ключові слова: логіка, предикат, логічний наслідок, секвенційне числення.

Київський національний університет імені
Тараса Шевченка, пр. Глушкова 4д, м. Київ, 03680
e-mail: ttp@unicyb.kiev.ua

M.S. Nikitchenko, Dr. Sci. (Phys.-Math.), Prof.,
S.S. Shkilniak, Dr. Sci. (Phys.-Math.).

**Composition-nominative logics
with special variable definedness
predicates**

Pure first-order composition-nominative logics extended with special predicates are investigated. These predicates indicate whether a variable is defined (has a value). Semantic properties of logics related with such predicates are studied. For logics of partial single-valued, total multi-valued, and partial multi-valued predicates sequent calculi are constructed. Their soundness and completeness are proved.

Key words: logic, predicate, logical consequence, sequent calculus.

Taras Shevchenko National University of Kyiv,
4d Glushkova str., Kyiv, 03680
e-mail: ttp@unicyb.kiev.ua

Статтю представив д. ф.-м. н., проф. Буй Д.Б.

Поняття і методи математичної логіки знаходять широке й ефективне застосування в інформатиці й програмуванні (див., напр., [1]). В основі різноманітних логічних систем, які використовуються в програмуванні, зазвичай лежить апарат класичної логіки предикатів. Проте ця логіка має принципові обмеження (див., напр., [2]), що ускладнює її застосування. Тому на перший план висувається задача побудови нових, програмно-орієнтованих логічних формалізмів. Природною основою такої побудови є спільний для логіки й програмування композиційно-номінативний підхід. Логіки, збудовані на його основі, названо композиційно-номінативними (КНЛ). Ці логіки базуються на загальних класах часткових відображень, заданих на довільних наборах іменованих значень. Такі відображення названі квазіарними.

Для КНЛ квазіарних предикатів важливим є те, що значення предиката $P(d)$ може бути різним залежно від того, входить чи не входить до d компонента з певним предметним іменем. Це веде до того, що для логіки квазіарних предикатів вже невірні [2, 3] деякі важливі закони класичної логіки. Тому при інтерпретаціях формул варто явно вказувати означені та неозначені предметні

імена. Для цього використаємо спеціальні 0-арні композиції – параметризовані за предметними іменами предикати-індикатори εz , які визначають наявність в даних компоненти з відповідним іменем (змінною) z , тобто наявність значення для z .

Метою цієї роботи є дослідження чистих першопорядкових КНЛ (ЧКНЛ) з виділеними предикатами-індикаторами, які називатимемо ε -ЧКНЛ. Розглянуто семантичні властивості ε -ЧКНЛ часткових однозначних, тотальних неоднозначних та часткових неоднозначних предикатів. Для різних відношень логічного наслідку таких логік побудовано числення секвенційного типу, доведено коректність та повноту цих числень

Поняття, які тут не визначаються, тлумачимо в сенсі робіт [2, 3, 4].

**1. Предикати-індикатори наявності значення
для предметних імен та їх властивості**

Предикати-індикатори εz наявності значення для відповідних предметних імен z задаємо так:

$$T(\varepsilon z) = \{d \mid d(z) \uparrow\} = \{d \in {}^V A \mid z \notin \text{asn}(d)\};$$

$$F(\varepsilon z) = \{d \mid d(z) \downarrow\} = \{d \in {}^V A \mid z \in \text{asn}(d)\}.$$

Такі εz не є монотонними і не є антитонними.

Розглянемо семантичні властивості предикатів-індикаторів. Надалі тотожно істинний предикат позначаємо як Т, тотожно хибний – як F.

Наведемо властивості предикатів εz , пов'язані з реномінаціями:

$$R_{\bar{y},u}^{\bar{x},z}(\varepsilon z) = \varepsilon u; \text{ зокрема, } R_u^z(\varepsilon z) = \varepsilon u;$$

$$R_{\bar{y}}^{\bar{x}}(\varepsilon z) = \varepsilon z, \text{ якщо } z \notin \{\bar{x}\};$$

$$\text{зокрема, } R_y^x(\varepsilon z) = \varepsilon z.$$

Наведемо властивості предикатів εz , пов'язані з композиціями квантифікації.

$$T(\exists x \varepsilon x) = T(\forall x \varepsilon x) = F(\exists x \neg \varepsilon x) = F(\forall x \neg \varepsilon x) = \emptyset;$$

$$F(\forall x \varepsilon x) = F(\exists x \varepsilon x) = T(\exists x \neg \varepsilon x) = T(\forall x \neg \varepsilon x) = {}^V A.$$

Таким чином, отримуємо:

$$\exists x \varepsilon x = F, \forall x \varepsilon x = F, \exists x \neg \varepsilon x = T, \forall x \neg \varepsilon x = T.$$

Для випадку різних імен квантора та предиката-індикатора маємо:

$$\exists x \varepsilon z = \forall x \varepsilon z = \varepsilon z; \exists x \neg \varepsilon z = \forall x \neg \varepsilon z = \neg \varepsilon z.$$

$$\text{Маємо: } T(\exists y(\varepsilon y \& P)) = T(\forall y(\varepsilon y \& P)) = \emptyset;$$

$$F(\exists y(\varepsilon y \& P)) = F(\forall y(\varepsilon y \& P)) = {}^V A;$$

$$T(\exists y(\neg \varepsilon y \vee P)) = T(\forall y(\neg \varepsilon y \vee P)) = {}^V A;$$

$$F(\exists y(\neg \varepsilon y \vee P)) = F(\forall y(\neg \varepsilon y \vee P)) = \emptyset.$$

Звідси такі властивості:

$$\exists y(\varepsilon y \& P) = \forall y(\varepsilon y \& P) = F;$$

$$\exists y(\neg \varepsilon y \vee P) = \forall y(\neg \varepsilon y \vee P) = T.$$

Подібним чином отримуємо:

$$\exists y(\varepsilon y \vee P) = \exists y(\neg \varepsilon y \& P) = \exists y P;$$

$$\forall y(\varepsilon y \vee P) = \forall y(\neg \varepsilon y \& P) = \forall y P.$$

Далі маємо (тут враховуємо, що x строго неістотне для εy та $\neg \varepsilon y$):

$$\exists x(P \vee \varepsilon y) = \exists x P \vee \exists x \varepsilon y = \exists x P \vee \varepsilon y;$$

$$\exists x(P \vee \neg \varepsilon y) = \exists x P \vee \exists x \neg \varepsilon y = \exists x P \vee \neg \varepsilon y;$$

$$\exists x(P \& \varepsilon y) = \exists x P \& \exists x \varepsilon y = \exists x P \& \varepsilon y;$$

$$\exists x(P \& \neg \varepsilon y) = \exists x P \& \exists x \neg \varepsilon y = \exists x P \& \neg \varepsilon y;$$

$$\forall x(P \vee \varepsilon y) = \forall x P \vee \forall x \varepsilon y = \forall x P \vee \varepsilon y;$$

$$\forall x(P \vee \neg \varepsilon y) = \forall x P \vee \forall x \neg \varepsilon y = \forall x P \vee \neg \varepsilon y;$$

$$\forall x(P \& \varepsilon y) = \forall x P \& \exists x \varepsilon y = \forall x P \& \varepsilon y;$$

$$\forall x(P \& \neg \varepsilon y) = \forall x P \& \forall x \neg \varepsilon y = \forall x P \& \neg \varepsilon y.$$

Семантичними моделями ε -ЧКНЛ є композиційні системи квазіарних предикатів кванторного рівня вигляду $({}^V A, Pr^A, C)$, де C задана множиною базових композицій $\{\neg, \vee, R_{\bar{x}}^{\bar{y}}, \exists x, \varepsilon z\}$. Така композиційна система задає неокласичну алгебру (алгебраїчну систему) даних (A, Pr^A) та композиційну алгебру предикатів (Pr^A, C) . Побудова композиційної системи визначає мову логіки: терми алгебри предикатів трактуємо як формули мови.

Зауважимо, що задання ε -ЧКНЛ можна трактувати як виділення окремого підрівня ЧКНЛ із базовими композиціями $\neg, \vee, R_{\bar{x}}^{\bar{y}}, \exists x, \varepsilon z$.

Враховуючи, що предикати-індикатори немонотонні та неантитонні, поняття ε -ЧКНЛ не розглядаємо для логік еквітонних предикатів та логік антитонних предикатів.

Алфавіт мови: множина V предметних імен; множина Ps предикатних символів (ПС) – сигнатура мови; символи базових композицій $\neg, \vee, R_{\bar{x}}^{\bar{y}}, \exists x, \varepsilon z$.

Множина Fr формул мови визначається так:

$$- Ps \subseteq Fr; \{\varepsilon z \mid z \in V\} \subseteq Fr;$$

$$- \Phi, \Psi \in Fr \Rightarrow \neg \Phi, \vee \Phi \Psi, R_{\bar{x}}^{\bar{y}} \Phi, \exists x \Phi \in Fr.$$

Інтерпретуємо мову на композиційних системах $({}^V A, Pr^A, C)$. Відображення інтерпретації формул $J : Fr \rightarrow Pr^A$ визначаємо на основі тотального однозначного відображення $I : Ps \rightarrow Pr^A$, яке позначає символами множини Ps базові предикати:

$$J(p) = I(p) \text{ для кожного } p \in Ps,$$

$$J(\varepsilon z) = \varepsilon z \text{ для кожного } z \in V;$$

$$J(\neg \Phi) = \neg(J(\Phi)), J(\vee \Phi \Psi) = \vee(J(\Phi), J(\Psi)),$$

$$J(R_{\bar{x}}^{\bar{y}} \Phi) = R_{\bar{x}}^{\bar{y}}(J(\Phi)), J(\exists x \Phi) = \exists x(J(\Phi)).$$

Відображення I пов'язує алгебру даних (A, Pr) із мовою ε -ЧКНЛ. Отримуємо об'єкт $((A, Pr^A), I)$ – алгебраїчну систему з доданою сигнатурою, яка визначає композиційну систему $({}^V A, Pr^A, C)$.

Алгебраїчні системи з доданою сигнатурою є інтегрованими семантичними моделями, які пов'язують мови КНЛ із алгебрами даних. Називаємо їх моделями мови та позначаємо $A = (A, I)$.

Предикат $J(\Phi)$ далі позначаємо Φ_A

Символи εz в кожній моделі мови A завжди інтерпретуються однаково, як відповідні предикати-індикатори εz . Тому для предикатів-індикаторів надалі будемо опускати індекс, який позначає модель мови (із контексту буде зрозуміло, це предикат чи предикатний символ).

Наведені вище властивості для предикатів можна записати як семантичні властивості формул відповідної мови ε -ЧКНЛ:

$$\exists x(\neg \varepsilon x \vee \Phi)_A = \exists x(\varepsilon x \rightarrow \Phi)_A = T;$$

$$\exists x \neg(\neg \varepsilon x \vee \Phi)_A = \exists x \neg(\varepsilon x \rightarrow \Phi)_A = F;$$

$$\forall x(\neg \varepsilon x \vee \Phi)_A = \forall x(\varepsilon x \rightarrow \Phi)_A = T;$$

$$\exists x(\varepsilon x \& \Phi)_A = \forall x(\varepsilon x \& \Phi)_A = F;$$

$$\exists x(\varepsilon x \vee \Phi)_A = \exists x(\neg \varepsilon x \& \Phi)_A = \exists x \Phi_A;$$

$$\exists x \neg(\varepsilon x \vee \Phi)_A = \exists x \neg(\neg \varepsilon x \& \Phi)_A = \exists x \neg \Phi_A;$$

$$\forall x(\varepsilon x \vee \Phi)_A = \forall x(\neg \varepsilon x \& \Phi)_A = \forall x \Phi_A;$$

$$\exists x(\Phi \vee \varepsilon y)_A = \exists x \Phi_A \vee \varepsilon y;$$

$$\exists x(\Phi \vee \neg \varepsilon y)_A = \exists x \Phi_A \vee \neg \varepsilon y;$$

$$\exists x \neg(\Phi \vee \varepsilon y)_A = \exists x \neg \Phi_A \& \neg \varepsilon y;$$

$$\exists x \neg(\Phi \vee \neg \varepsilon y)_A = \exists x \neg \Phi_A \& \varepsilon y;$$

$$\exists x(\Phi \& \varepsilon y)_A = \exists x \Phi_A \& \varepsilon y;$$

$$\exists x(\Phi \& \neg \varepsilon y)_A = \exists x \Phi_A \& \neg \varepsilon y;$$

$$\forall x(\Phi \vee \varepsilon y)_A = \forall x \Phi_A \vee \varepsilon y;$$

$$\forall x(\Phi \vee \neg \epsilon y)_A = \forall x \Phi_A \vee \neg \epsilon y;$$

$$\forall x(\Phi \& \epsilon y)_A = \forall x \Phi_A \& \epsilon y;$$

$$\forall x(\Phi \& \neg \epsilon y)_A = \forall x \Phi_A \& \neg \epsilon y.$$

Формули, які задають константні предикати T та F, назовемо *константними*.

Константні формули вигляду $\exists x \neg \epsilon x$, $\neg \exists x \epsilon x$, $\exists x(\neg \epsilon x \vee \Phi)$, $\neg \exists x \neg(\neg \epsilon x \vee \Phi)$ задають T; назовемо їх ϵT -формулами.

Константні формули вигляду $\exists x \epsilon x$, $\neg \exists x \neg \epsilon x$, $\exists x \neg(\neg \epsilon x \vee \Phi)$, $\neg \exists x(\neg \epsilon x \vee \Phi)$ задають F; назовемо їх ϵF -формулами.

ϵT -формули та ϵF -формули назовемо *виділеними константними ϵ -формулами*.

Випишемо тепер основні властивості формул із предикатами-індикаторами, записані для базових композицій $\neg$, $\vee$, $R_{\bar{y},u}^{\bar{x},z}$, $\exists x$:

$$- R_{\bar{y},u}^{\bar{x},z}(\epsilon z) \sim_{TF} \epsilon u; \text{ зокрема, } R_u^z(\epsilon z) \sim_{TF} \epsilon u;$$

$$- R_{\bar{y}}^{\bar{x}}(\epsilon z) \sim_{TF} \epsilon z, \text{ якщо } z \notin \{\bar{x}\};$$

$$- \exists x \epsilon z \sim_{TF} \epsilon z; \exists x \neg \epsilon z \sim_{TF} \neg \epsilon z;$$

$$- \exists x(\epsilon x \vee \Phi) \sim_{TF} \exists x \Phi;$$

$$- \exists x \neg(\epsilon x \vee \Phi) \sim_{TF} \exists x \neg \Phi;$$

$$- \exists x(\Phi \vee \epsilon y) \sim_{TF} \exists x \Phi \vee \epsilon y;$$

$$- \exists x(\Phi \vee \neg \epsilon y) \sim_{TF} \exists x \Phi \vee \neg \epsilon y.$$

Властивості формул із предикатами-індикаторами індукують відповідні властивості відношень логічного наслідку для множин формул.

Надалі, якщо інше не зазначено, $\models$ позначає:

– $\models_{TF}$ для загальної семантики;

– одне із $\models_{Cl}$, $\models_T$, $\models_F$, $\models_{TF}$ для неокласичної семантики та одне із $\models_{Cm}$, $\models_T$, $\models_F$, $\models_{TF}$ для пересиченої семантики.

Із визначень ϵu та цих відношень отримуємо:

$$\neg \epsilon) \neg \epsilon u, \Gamma \models \Delta \Leftrightarrow \Gamma \models \Delta, \epsilon u;$$

$$\Gamma \models \Delta, \neg \epsilon u \Leftrightarrow \epsilon u, \Gamma \models \Delta.$$

Властивості формул із предикатами-індикаторами індукують відповідні властивості відношень $\models_{TF}$, $\models_T$, $\models_F$, $\models_{Cl}$, $\models_{Cm}$:

$$\exists \epsilon) \exists x \epsilon z, \Gamma \models \Delta \Leftrightarrow \epsilon z, \Gamma \models \Delta \text{ та}$$

$$\Gamma \models \Delta, \exists x \epsilon z \Leftrightarrow \Gamma \models \Delta, \epsilon z.$$

$$\exists \neg \epsilon) \exists x \neg \epsilon z, \Gamma \models \Delta \Leftrightarrow \neg \epsilon z, \Gamma \models \Delta \text{ та}$$

$$\Gamma \models \Delta, \exists x \neg \epsilon z \Leftrightarrow \Gamma \models \Delta, \neg \epsilon z.$$

$$\neg \exists \epsilon) \neg \exists x \epsilon z, \Gamma \models \Delta \Leftrightarrow \neg \epsilon z, \Gamma \models \Delta \text{ та}$$

$$\Gamma \models \Delta, \neg \exists x \epsilon z \Leftrightarrow \Gamma \models \Delta, \neg \epsilon z.$$

$$\neg \exists \neg \epsilon) \neg \exists x \neg \epsilon z, \Gamma \models \Delta \Leftrightarrow \epsilon z, \Gamma \models \Delta \text{ та}$$

$$\Gamma \models \Delta, \neg \exists x \neg \epsilon z \Leftrightarrow \Gamma \models \Delta, \epsilon z.$$

$$R \epsilon) R_{\bar{y}}^{\bar{x}}(\epsilon z), \Gamma \models \Delta \Leftrightarrow \epsilon z, \Gamma \models \Delta \text{ та}$$

$$\Gamma \models \Delta, R_{\bar{y}}^{\bar{x}}(\epsilon z) \Leftrightarrow \Gamma \models \Delta, \epsilon z.$$

$$\neg R \epsilon) \neg R_{\bar{y}}^{\bar{x}}(\epsilon z), \Gamma \models \Delta \Leftrightarrow \neg \epsilon z, \Gamma \models \Delta \text{ та}$$

$$\Gamma \models \Delta, \neg R_{\bar{y}}^{\bar{x}}(\epsilon z) \Leftrightarrow \Gamma \models \Delta, \neg \epsilon z.$$

Для $R \epsilon$ та $\neg R \epsilon$ умова: $z \notin \{\bar{x}\}$.

$$R \epsilon \vee) R_{\bar{y},u}^{\bar{x},z}(\epsilon z), \Gamma \models \Delta \Leftrightarrow \epsilon u, \Gamma \models \Delta \text{ та}$$

$$\Gamma \models \Delta, R_{\bar{y},u}^{\bar{x},z}(\epsilon z) \Leftrightarrow \Gamma \models \Delta, \epsilon u.$$

$$\neg R \epsilon \vee) \neg R_{\bar{y},u}^{\bar{x},z}(\epsilon z), \Gamma \models \Delta \Leftrightarrow \neg \epsilon u, \Gamma \models \Delta \text{ та}$$

$$\Gamma \models \Delta, \neg R_{\bar{y},u}^{\bar{x},z}(\epsilon z) \Leftrightarrow \Gamma \models \Delta, \neg \epsilon u.$$

Додаткові властивості:

$$\exists \vee \epsilon) \exists x(\epsilon x \vee \Phi), \Gamma \models \Delta \Leftrightarrow \exists x \Phi, \Gamma \models \Delta \text{ та}$$

$$\Gamma \models \Delta, \exists x(\epsilon x \vee \Phi) \Leftrightarrow \Gamma \models \Delta, \exists x \Phi.$$

$$\neg \exists \vee \epsilon) \neg \exists x(\epsilon x \vee \Phi), \Gamma \models \Delta \Leftrightarrow \neg \exists x \Phi, \Gamma \models \Delta \text{ та}$$

$$\Gamma \models \Delta, \neg \exists x(\epsilon x \vee \Phi) \Leftrightarrow \Gamma \models \Delta, \neg \exists x \Phi.$$

$$\exists \neg \vee \epsilon) \exists x \neg(\epsilon x \vee \Phi), \Gamma \models \Delta \Leftrightarrow \exists x \neg \Phi, \Gamma \models \Delta \text{ та}$$

$$\Gamma \models \Delta, \exists x \neg(\epsilon x \vee \Phi) \Leftrightarrow \Gamma \models \Delta, \exists x \neg \Phi.$$

$$\neg \exists \neg \vee \epsilon) \neg \exists x \neg(\epsilon x \vee \Phi), \Gamma \models \Delta \Leftrightarrow \neg \exists x \neg \Phi, \Gamma \models \Delta$$

$$\text{та } \Gamma \models \Delta, \neg \exists x \neg(\epsilon x \vee \Phi) \Leftrightarrow \Gamma \models \Delta, \neg \exists x \neg \Phi.$$

Наведемо пов'язані з виділеними константними ϵ -формулами умови, які гарантують наявність логічного наслідку (випливають із того, що $\exists x \epsilon x = \neg \exists x \neg \epsilon x = F$ та $\neg \exists x \epsilon x = \exists x \neg \epsilon x = T$):

$$C \epsilon_{\neg}) \exists x \epsilon x, \Gamma \models \Delta; \neg \exists x \neg \epsilon x, \Gamma \models \Delta;$$

$$C \epsilon_{\neg}) \Gamma \models \Delta, \neg \exists x \epsilon x; \Gamma \models \Delta, \exists x \neg \epsilon x.$$

Додаткові пов'язані з виділеними константними ϵ -формулами умови, які гарантують наявність логічного наслідку (вони випливають із того, що $(\exists x \neg(\neg \epsilon x \vee \Phi) = \neg \exists x(\neg \epsilon x \vee \Phi) = F$ та $\exists x(\neg \epsilon x \vee \Phi)_A = \neg \exists x(\neg \epsilon x \vee \Phi) = T$):

$$C \Phi \epsilon_{\neg}) \exists x \neg(\neg \epsilon x \vee \Phi), \Gamma \models \Delta; \neg \exists x(\neg \epsilon x \vee \Phi), \Gamma \models \Delta;$$

$$C \Phi \epsilon_{\neg}) \Gamma \models \Delta, \exists x(\neg \epsilon x \vee \Phi); \Gamma \models \Delta, \neg \exists x \neg(\neg \epsilon x \vee \Phi).$$

2. Секвенційні числення ϵ -ЧКНЛ

Побудуємо секвенційні числення ϵ -ЧКНЛ, які формалізують відношення $\models_{TF}$, $\models_T$, $\models_F$, $\models_{Cl}$, $\models_{Cm}$ у відповідних семантиках. Зауважимо, що для загальної семантики $\models_T$, $\models_F$, $\models_{TF}$ збігаються.

Наведені вище умови гарантованого логічного наслідку індукують пов'язані з предикатами-індикаторами додаткові умови замкненості секвенції в численнях ϵ -ЧКНЛ. Вони задаються наявністю в секвенції виділених константних ϵ -формул. Назвемо їх ϵ -умовами замкненості секвенції.

Основні ϵ -умови замкненості секвенції Ξ :

$$C \epsilon) \vdash \exists x \epsilon x \in \Xi;$$

$$C \neg \epsilon) \vdash \exists x \neg \epsilon x \in \Xi;$$

$$\neg C \epsilon) \vdash \neg \exists x \epsilon x \in \Xi;$$

$$\neg C \neg \epsilon) \vdash \neg \exists x \neg \epsilon x \in \Xi.$$

Додаткові (похідні) ϵ -умови замкненості Ξ :

$$C \Phi \epsilon) \vdash \exists x(\neg \epsilon x \vee \Phi) \in \Xi;$$

$$C \neg \Phi \epsilon) \vdash \exists x \neg(\neg \epsilon x \vee \Phi) \in \Xi;$$

$$\neg C\Phi\epsilon \vdash \neg \exists x(\neg \epsilon x \vee \Phi) \in \Xi;$$

$$\neg C\Phi \neg \epsilon \vdash \neg \exists x \neg (\neg \epsilon x \vee \Phi) \in \Xi.$$

Наведемо секвенційні форми спрощення, пов'язані з предикатами-індикаторами. Вони індукуються відповідними властивостями відношень логічного наслідку для множин формул із предикатами-індикаторами. Назвемо їх ϵ -формами.

$$\vdash \neg \epsilon \frac{\neg \epsilon x, \Sigma}{\neg \epsilon x, \Sigma}; \quad \vdash \neg \epsilon \frac{\neg \epsilon x, \Sigma}{\neg \epsilon x, \Sigma};$$

$$\vdash \exists_s \epsilon \frac{\neg \epsilon z, \Sigma}{\neg \exists x \epsilon z, \Sigma}; \quad \vdash \exists_s \epsilon \frac{\neg \epsilon z, \Sigma}{\neg \exists x \epsilon z, \Sigma};$$

$$\vdash \neg \exists_s \epsilon \frac{\neg \epsilon z, \Sigma}{\neg \neg \exists x \epsilon z, \Sigma}; \quad \vdash \neg \exists_s \epsilon \frac{\neg \epsilon z, \Sigma}{\neg \neg \exists x \epsilon z, \Sigma};$$

$$\vdash R\epsilon \frac{\neg \epsilon z, \Sigma}{\neg R_y^x(\epsilon z), \Sigma}; \quad \vdash R\epsilon \frac{\neg \epsilon z, \Sigma}{\neg R_y^x(\epsilon z), \Sigma};$$

$$\vdash \neg R\epsilon \frac{\neg \epsilon z, \Sigma}{\neg \neg R_y^x(\epsilon z), \Sigma}; \quad \vdash \neg R\epsilon \frac{\neg \epsilon z, \Sigma}{\neg \neg R_y^x(\epsilon z), \Sigma}.$$

Для форм типу $R\epsilon$ та $\neg R\epsilon$ умова: $z \notin \{\bar{x}\}$.

$$\vdash R\epsilon v \frac{\neg \epsilon u, \Sigma}{\neg R_{y,u}^{\bar{x},z}(\epsilon z), \Sigma}; \quad \vdash R\epsilon v \frac{\neg \epsilon u, \Sigma}{\neg R_{y,u}^{\bar{x},z}(\epsilon z), \Sigma};$$

$$\vdash \neg R\epsilon v \frac{\neg \epsilon u, \Sigma}{\neg \neg R_{y,u}^{\bar{x},z}(\epsilon z), \Sigma}; \quad \vdash \neg R\epsilon v \frac{\neg \epsilon u, \Sigma}{\neg \neg R_{y,u}^{\bar{x},z}(\epsilon z), \Sigma};$$

$$\vdash \exists v \epsilon \frac{\neg \exists x \Phi, \Sigma}{\neg \exists x(\epsilon x \vee \Phi), \Sigma}; \quad \vdash \exists v \epsilon \frac{\neg \exists x \Phi, \Sigma}{\neg \exists x(\epsilon x \vee \Phi), \Sigma};$$

$$\vdash \neg \exists v \epsilon \frac{\neg \neg \exists x \Phi, \Sigma}{\neg \neg \exists x(\epsilon x \vee \Phi), \Sigma}; \quad \vdash \neg \exists v \epsilon \frac{\neg \neg \exists x \Phi, \Sigma}{\neg \neg \exists x(\epsilon x \vee \Phi), \Sigma};$$

$$\vdash \exists \neg v \epsilon \frac{\neg \exists x \neg \Phi, \Sigma}{\neg \exists x \neg (\epsilon x \vee \Phi), \Sigma}; \quad \vdash \exists \neg v \epsilon \frac{\neg \exists x \neg \Phi, \Sigma}{\neg \exists x \neg (\epsilon x \vee \Phi), \Sigma};$$

$$\vdash \neg \exists \neg v \epsilon \frac{\neg \neg \exists x \neg \Phi, \Sigma}{\neg \neg \exists x \neg (\epsilon x \vee \Phi), \Sigma}; \quad \vdash \neg \exists \neg v \epsilon \frac{\neg \neg \exists x \neg \Phi, \Sigma}{\neg \neg \exists x \neg (\epsilon x \vee \Phi), \Sigma}.$$

Зауважимо, що форми типів $\exists v \epsilon$, $\neg \exists v \epsilon$, $\exists \neg v \epsilon$, $\neg \exists \neg v \epsilon$ є додатковими (похідними); те ж можна сказати про форми типів $\exists_s \epsilon$ та $\neg \exists_s \epsilon$.

Розглянемо секвенційні числення ϵ -ЧКНЛ.

QEG-числення формалізує відношення $\models_{TF}$ для випадку загальної семантики.

Умова замкненості секвенції:

$$C \vee \text{Un}C \vee C\epsilon \vee C\neg\epsilon \vee \neg C\epsilon \vee \neg C\neg\epsilon \vee$$

$$\vee C\Phi\epsilon \vee C\neg\Phi\epsilon \vee \neg C\Phi\epsilon \vee \neg C\neg\Phi\epsilon.$$

Тут $C\epsilon \vee C\neg\epsilon \vee \neg C\epsilon \vee \neg C\neg\epsilon \vee C\Phi\epsilon \vee C\neg\Phi\epsilon \vee \neg C\Phi\epsilon \vee \neg C\neg\Phi\epsilon$ – додаткова пов'язана з виділеними константними ϵ -формулами умова замкненості секвенції. Зауважимо, що тут умова $C\Phi\epsilon \vee C\neg\Phi\epsilon \vee \neg C\Phi\epsilon \vee \neg C\neg\Phi\epsilon$ не є обов'язковою але її застосування спрощує виведення.

Базові секвенційні форми – це форми числення QSG (див. [4]): $\vdash RT, \neg RT, \vdash \neg RT, \neg \neg RT, \vdash \Phi N, \neg \Phi N, \vdash \neg \Phi N, \neg \neg \Phi N, \vdash R\exists R, \neg R\exists R, \vdash \neg R\exists R, \neg \neg R\exists R, \vdash R\exists p, \neg R\exists p, \vdash \neg R\exists p, \neg \neg R\exists p, \vdash \neg \neg, \neg \neg \neg, \vdash \vee, \neg \vee, \vdash \neg \vee, \neg \neg \vee, \vdash RR, \neg RR, \vdash \neg RR, \neg \neg RR, \vdash R\neg, \neg R\neg, \vdash \neg R\neg, \neg \neg R\neg, \vdash R\vee, \neg R\vee, \vdash \neg R\vee, \neg \neg R\vee, \vdash \exists \epsilon, \neg \exists \epsilon, \vdash \neg \exists \epsilon, \neg \neg \exists \epsilon, \vdash \exists f, \neg \exists f, \vdash \neg \exists f, \neg \neg \exists f, \vdash \exists v, \neg \exists v, \vdash \neg \exists v, \neg \neg \exists v, \vdash \exists d, \neg \exists d, \vdash \neg \exists d, \neg \neg \exists d$.

До них додаємо пов'язані з предикатами-індикаторами ϵ -форми (вони наведені вище).

QEL-числення формалізує відношення $\models_T$ (неокласична семантика) та $\models_F$ (пересичена семантика). Умова замкненості секвенції:

$$C \vee CL \vee \text{Un}C \vee C\epsilon \vee C\neg\epsilon \vee \neg C\epsilon \vee \neg C\neg\epsilon \vee$$

$$\vee C\Phi\epsilon \vee C\neg\Phi\epsilon \vee \neg C\Phi\epsilon \vee \neg C\neg\Phi\epsilon.$$

Базові секвенційні форми та ϵ -форми такі ж, як для числення QEG.

QER-числення формалізує відношення $\models_F$ (неокласична семантика) та $\models_T$ (пересичена семантика). Умова замкненості секвенції:

$$C \vee CR \vee \text{Un}C \vee C\epsilon \vee C\neg\epsilon \vee \neg C\epsilon \vee \neg C\neg\epsilon \vee$$

$$\vee C\Phi\epsilon \vee C\neg\Phi\epsilon \vee \neg C\Phi\epsilon \vee \neg C\neg\Phi\epsilon.$$

Базові секвенційні форми та ϵ -форми такі ж, як для числення QEG.

QELR-числення формалізує відношення $\models_{TF}$ для випадків неокласичної семантики та пересиченої семантики. Умова замкненості секвенції:

$$C \vee CLR \vee \text{Un}C \vee C\epsilon \vee C\neg\epsilon \vee \neg C\epsilon \vee \neg C\neg\epsilon \vee$$

$$\vee C\Phi\epsilon \vee C\neg\Phi\epsilon \vee \neg C\Phi\epsilon \vee \neg C\neg\Phi\epsilon.$$

Базові секвенційні форми та ϵ -форми такі ж, як для числення QEG.

QEC-числення формалізує відношення $\models_{CI}$ (неокласична семантика) та $\models_{Cm}$ (пересичена семантика). Умова замкненості секвенції:

$$C \vee \text{Un}C \vee C\epsilon \vee C\neg\epsilon \vee C\Phi\epsilon \vee C\neg\Phi\epsilon.$$

Тут $C\epsilon \vee C\neg\epsilon \vee C\Phi\epsilon \vee C\neg\Phi\epsilon$ – додаткова умова замкненості секвенції. Зауважимо, що тут умова $C\Phi\epsilon \vee C\neg\Phi\epsilon$ не є обов'язковою.

Базові секвенційні форми – це форми числення QSC [4]: $\vdash RT, \neg RT, \vdash \Phi N, \neg \Phi N, \vdash R\exists R, \neg R\exists R, \vdash R\exists p, \neg R\exists p, \vdash \neg, \neg \neg, \vdash \vee, \neg \vee, \vdash RR, \neg RR, \vdash R\neg, \neg R\neg, \vdash R\vee, \neg R\vee, \vdash \exists \epsilon, \neg \exists \epsilon, \vdash \exists f, \neg \exists f, \vdash \exists v, \neg \exists v, \vdash \exists d, \neg \exists d$.

До них додаємо наступні ϵ -форми: $\vdash \neg \epsilon, \neg \neg \epsilon, \vdash \exists \epsilon, \neg \exists \epsilon, \vdash R\epsilon, \neg R\epsilon, \vdash R\epsilon v, \neg R\epsilon v, \vdash \exists v \epsilon, \neg \exists v \epsilon, \vdash \exists \neg v \epsilon, \neg \exists \neg v \epsilon$.

Теорема 1. Нехай $\frac{\vdash \Lambda \neg K}{\vdash \Gamma \neg \Delta}$ та $\frac{\vdash \Lambda \neg K \quad \vdash X \neg Z}{\vdash \Gamma \neg \Delta}$ –

базові секвенційні форми. Тоді маємо:

- 1) $\Lambda \models K \Rightarrow \Gamma \models \Delta$; $\Lambda \models K$ та $X \models Z \Rightarrow \Gamma \models \Delta$;
- 2) $\Gamma \models \Delta \Rightarrow \Lambda \models K$; $\Gamma \models \Delta \Rightarrow \Lambda \models K$ або $X \models Z$.

Наведемо спектр побудованих секвенційних числень ε -ЧКНЛ (табл. 1).

Таблиця 1. Секвенційні числення ε -ЧКНЛ.

	$\models_{Cl}$	$\models_{Cm}$	$\models_T$	$\models_F$	$\models_{TF}$
Нео	$Q\in C$	–	$Q\in L$	$Q\in S$	$Q\in LR$
Пер	–	$Q\in C$	$Q\in R$	$Q\in L$	$Q\in LR$
Заг	–	–	$Q\in G$	$Q\in G$	$Q\in G$

Тут скорочення "Нео", "Пер", "Заг" відповідно означають: "неокласична семантика", "пересичена семантика", "загальна семантика".

3. Коректність та повнота секвенційних числень

Процедура побудови секвенційного дерева для числень ε -ЧКНЛ в цілому така ж, як процедура побудови дерева для відповідних числень ЧКНЛ. При цьому додатково враховуємо наявність ε -форм, а виділені константні ε -формули трактуються як примітивні.

Якщо процедура побудови дерева для секвенції Σ завершена позитивно, то маємо скінченне замкнене дерево, всі його вершини – замкнені секвенції. Якщо процедура завершена негативно (маємо скінченне незамкнене дерево) або процедура не завершується (будоване дерево нескінченне), то у дереві існує незамкнений шлях $\wp$, всі його вершини – незамкнені секвенції. Кожна з формул Σ зустрінеться на $\wp$ і стане доступною.

Теорема 2 (коректності). Секвенція $\vdash \Gamma \rightarrow \Delta$ вивідна в численні $\beta \Rightarrow \Gamma \models \Delta$ в семантиці σ .

Назва β числення – на перетині стовпця $\models$ та рядка σ (табл. 1).

Якщо $\vdash \Gamma \rightarrow \Delta$ вивідна, то для неї побудоване замкнене секвенційне дерево. Для кожної його вершини $\vdash \Lambda \rightarrow K$ маємо $\Lambda \models K$ (для листів дерева це впливає з їх замкненості, а збереження секвенційними формами відповідних відношень $\models_{Cl}$, $\models_{Cm}$, $\models_T$, $\models_F$, $\models_{TF}$ впливає із теореми 1). Зокрема, це вірно для кореня дерева $\vdash \Gamma \rightarrow \Delta$, тобто $\Gamma \models_* \Delta$.

Повнота побудованих числень впливає із відповідних теорем про існування контрмоделі для множини формул незамкненого шляху секвенційного дерева. Доведення цих теорем опирається на метод модельних множин (див., напр., [2, 6, 5]).

Побудова контрмоделей для числень ε -ЧКНЛ має враховувати наявність виділених константних ε -формул. Поряд із заданням значень базових предикатів та їх заперечень, для числень ε -ЧКНЛ задаємо фіксовані значення таких формул.

Нехай τ – одна з εT -формул, ξ – одна з εF -формул. Для кожної моделі мови $A = (A, I)$ та кожного $d \in {}^V A$ тоді маємо:

$$T(\tau_A) = {}^V A \text{ та } F(\tau_A) = \emptyset; \quad T(\xi_A) = \emptyset \text{ та } F(\xi_A) = {}^V A.$$

Теорема 3 (про контрмоделі для числення $Q\in G$). Нехай $\wp$ – незамкнений шлях у секвенційному дереві, збудованому для $\vdash \Gamma \rightarrow \Delta$, нехай H – множина всіх специфікованих формул секвенцій цього шляху. Тоді існують моделі мови $A = (A, I)$, $B = (A, I)$ та $\delta, \eta \in {}^V A$:

- 1) $\vdash \Phi \in H \Rightarrow \delta \in T(\Phi_A)$ та $\vdash \Phi \in H \Rightarrow \delta \notin F(\Phi_A)$;
- 2) $\vdash \Phi \in H \Rightarrow \eta \notin F(\Phi_B)$ та $\vdash \Phi \in H \Rightarrow \eta \in F(\Phi_B)$.

Такі пари (A, δ) та (B, η) назовемо *T-контрмоделлю* та *F-контрмоделлю* для секвенції $\vdash \Gamma \rightarrow \Delta$.

Для множини H формулюємо умови коректності та умови переходу. Це робимо так, як для G -модельної множини [4, 5], із додаванням умов коректності, пов'язаних з виділеними константними ε -формулами, та умов переходу, пов'язаних з ε -формами. Така множина H – εG -модельна.

За множиною H будуємо контрмоделі. Нехай $W = \{y \in nm(H) \mid \neg y \in H\}$. Візьмемо деяку множину A таку, що $|A| = |W|$, та деякі ін'єктивні $\delta, \eta \in {}^V A$ з $asn(\delta) = W$. Значення базових предикатів та їх заперечень на δ і η та на даних вигляду $r_{\bar{x}}^{\bar{v}}(\delta)$ і $r_{\bar{x}}^{\bar{v}}(\eta)$ задаємо так, як для QSG-числень (див. [4]). Окрім того, задамо значення εT -формул та εF -формул як константні предикати T та F .

Зауважимо, що для εT -формули τ неможливо $\vdash \tau \in H$ та для εF -формули ξ неможливо $\vdash \xi \in H$, адже H – це множина всіх специфікованих формул секвенцій *незамкненого* шляху, а $\vdash \tau \in \Xi$ та $\vdash \xi \in \Xi$ – це умови *замкненості* секвенції Ξ .

Таким чином, для εT -формул τ та εF -формул ξ маємо виконання умов теореми 3:

$$\begin{aligned} & \vdash \tau \in H \Rightarrow \delta \in T(\tau_A) = {}^V A; \quad \vdash \tau \in H \Rightarrow \eta \notin F(\tau_B) = \emptyset; \\ & \vdash \xi \in H \Rightarrow \delta \notin T(\xi_A) = \emptyset; \quad \vdash \xi \in H \Rightarrow \eta \in F(\xi_B) = {}^V A. \end{aligned}$$

Далі доведення традиційне: індукцією за складністю формули згідно з умовами переходу множини H (подібні доведення див. [4, 5]).

Теореми про контрмоделі для числень $Q\in L$, $Q\in R$, $Q\in LR$ формулюються і доводяться аналогічно теоремі 3. Для числення $Q\in C$ теорема про контрмоделі формулюється так, як для числення QSC .

Теорема 4 (про контрмоделі для числення $Q\in C$). Нехай $\wp$ – незамкнений шлях у секвенційному дереві, збудованому для $\vdash \Gamma \rightarrow \Delta$, нехай H – множина всіх специфікованих формул секвенцій цього шляху. Тоді існують моделі мови $A = (A, I)$, $B = (A, I)$ та $\delta, \eta \in {}^V A$:

- 1) $\vdash \Phi \in H \Rightarrow \delta \in T(\Phi_A)$ та $\vdash \Phi \in H \Rightarrow \delta \in F(\Phi_A)$;
- 2) $\vdash \Phi \in H \Rightarrow \eta \notin F(\Phi_B)$ та $\vdash \Phi \in H \Rightarrow \eta \notin T(\Phi_B)$.

Такі (A, δ) та (B, η) назвемо *CI-контрмоделлю* та *Ст-контрмоделлю* для секвенції $\vdash \Gamma \vdash \Delta$.

Для множини H умови коректності та умови переходу формулюються так, як для C -модельної множини [4, 5], із додаванням умов коректності, пов'язаних з виділеними константними ε -формулами, та умов переходу, пов'язаних з ε -формами. Таку множину H назвемо εC -модельною.

За множиною H будуються контрмоделі. Нехай $W = \{y \in nm(H) \mid \vdash y \in H\}$. Візьмемо деяку множину A таку, що $|A| = |W|$, та деякі ін'єктивні $\delta, \eta \in {}^V A$ з $asn(\delta) = W$. Значення базових предикатів на δ і η та на даних вигляду $r_{\bar{x}}^{\bar{y}}(\delta)$ і $r_{\bar{x}}^{\bar{y}}(\eta)$ задаємо так, як для QSC -числень (див. [4, 5]). Додатково задамо значення εT -формул та εF -формул як константні предикати T та F .

Для εT -формули τ неможливо $\vdash \tau \in H$ та для εF -формули ξ неможливо $\vdash \xi \in H$. Тому для εT -формул τ та εF -формул ξ виконані умови теореми 4: $\vdash \tau \in H \Rightarrow \delta \in T(\tau_A) = {}^V A$; $\vdash \tau \in H \Rightarrow \eta \notin F(\tau_B) = \emptyset$; $\vdash \xi \in H \Rightarrow \delta \in F(\xi_A) = {}^V A$; $\vdash \xi \in H \Rightarrow \eta \notin T(\xi_B) = \emptyset$.

Далі доведення ведемо індукцією за складністю формули згідно з умовами переходу множини H .

Для різних числень та логічних наслідків ε -ЧКНЛ теореми повноти формулюються однотипно. Вони доводяться традиційним чином (див., напр., [4, 5]) на основі теорем про контрмоделі.

Теорема 5 (повноти). $\Gamma \models \Delta$ в семантиці $\sigma \Rightarrow$ секвенція $\vdash \Gamma \vdash \Delta$ вивідна в численні β .

Назва β числення – на перетині стовпця $\models$ та рядка σ (табл. 1).

Наведемо для прикладу доведення для числення $Q\&G$ та відношення $\models_{TF}$ в загальній семантиці.

Нехай супротивне: $\Gamma \models_{TF} \Delta$ та $\vdash \Gamma \vdash \Delta$ невивідна. Тоді в дереві для $\vdash \Gamma \vdash \Delta$ існує незамкнений шлях. Нехай H – множина усіх специфікованих формул секвенцій цього шляху, тоді $\vdash \Gamma \vdash \Delta \subseteq H$. За теоремою 3 існують T -контрмодель (A, δ) та F -контрмодель (B, η) такі:

- $$\begin{aligned} \vdash \Phi \in H &\Rightarrow \delta \in T(\Phi_A) \text{ та } \vdash \Phi \in H \Rightarrow \delta \notin T(\Phi_A); \\ \vdash \Phi \in H &\Rightarrow \eta \notin F(\Phi_B) \text{ та } \vdash \Phi \in H \Rightarrow \eta \in F(\Phi_B). \end{aligned}$$

Можна брати як T -контрмодель, так і F -контрмодель. Для T -контрмоделі згідно з $\vdash \Gamma \vdash \Delta \subseteq H$ для всіх $\Phi \in \Gamma$ маємо $\delta \in T(\Phi_A)$, для всіх $\Psi \in \Delta$ маємо $\delta \notin T(\Psi_A)$. Звідси $\delta \in T(\Gamma_A)$ та $\delta \notin T(\Delta_A)$, тому невірнo $T(\Gamma_A) \subseteq T(\Delta_A)$, що заперечує $\Gamma_A \models_T \Delta$ та $\Gamma \models_{TF} \Delta$. Для F -контрмоделі згідно з $\vdash \Gamma \vdash \Delta \subseteq H$ маємо

$\eta \notin F(\Phi_B)$ для всіх $\Phi \in \Gamma$ та $\eta \in F(\Psi_B)$ для всіх $\Psi \in \Delta$. Звідси $\eta \notin F(\Gamma_B)$ та $\eta \in F(\Delta_B)$, тому невірнo $F(\Delta_B) \subseteq F(\Gamma_B)$, що заперечує $\Gamma_B \models_F \Delta$ та $\Gamma \models_{TF} \Delta$.

Подібним чином доводимо для відповідних відношень і семантик та числень $Q\&L$, $Q\&R$, $Q\&LR$ (див. табл. 1). Особливості вибору T -контрмоделі та F -контрмоделі для цих числень такі ж, як для числень QSL , QSR , $QSLR$ (див. [4]).

Висновки

В роботі досліджено чисті першопорядкові композиційно-номінативні логіки квазіарних предикатів, розширені спеціальними предикатами-індикаторами наявності значення для предметних імен (змінних). Такі логіки названі ε -ЧКНЛ. Розглянуто семантичні властивості ε -ЧКНЛ, пов'язані з предикатами-індикаторами, зокрема, властивості відношень логічного наслідку для множин формул. Для різних відношень логічного наслідку в ε -ЧКНЛ часткових однозначних, тотальних неоднозначних та часткових неоднозначних предикатів побудовано числення секвенційного типу. Для пропонованих числень доведено теореми коректності та повноти.

Список використаних джерел

1. *Handbook of Logic in Computer Science*. Edited by S. Abramsky, Dov M. Gabbay and T. S. E. Maibaum. – Oxford Univ. Press. – Vol. 1–5, 1993–2000.
2. *Nikitchenko M.S., Shkilniak S.S. Mathematical Logic and Theory of Algorithms*. – Kyiv: VPC Kyivskyi Universytet, 2008. – 528 p. (in Ukrainian).
3. *Nikitchenko M.S., Shkilniak S.S. Composition-nominative logics of quasiary predicates: semantic aspects* // Visnyk, Ser. Fiz.-Mat. Nauky, Kyiv Univ. im. Tarasa Shevchenka. – 2012. – № 4. – P. 165–172 (in Ukrainian).
4. *Shkilniak S.S. Sequent calculi for logics of partial and multi-valued quasi-ary predicates* // Visnyk, Ser. Fiz.-Mat. Nauky, Kyiv Univ. im. Tarasa Shevchenka. – 2012. – № 4. – P. 231–236 (in Ukrainian).
5. *Shkilniak S.S. Sequent Calculi of Composition Nominative Logics of Quasi-ary Predicates* // Problems in Programming. – 2012. – № 2–3. – P. 33–43 (in Ukrainian).

Надійшла до редколегії 09.09.2013

УДК 681.3

Олецький О.В.¹, к.т.н., доц.

Про оптимізацію структури веб-порталу на основі марковських процесів прийняття рішень

Обговорюється підхід до оптимізації структури гіпертекстових посилань на тематичному веб-порталі на основі застосування марковських процесів прийняття рішень. Пропонується формалізована модель, яка описує зв'язки між вузлами навігаційного графа та онтологією предметної області, спрямована на підвищення семантичної орієнтованості процесу навігації. Аналізуються можливості, пов'язані з застосуванням описаної методики в системах дистанційної освіти.

Ключові слова: навігаційний граф, навчання з підкріпленням, марковські процеси прийняття рішень, система e-learning.

¹Національний Університет “Києво-Могилянська Академія”, 04070, м.Київ, вул. Сковороди,2
e-mail: oletsky@ukr.net

O.V. Oletsky¹, PhD

On optimizing the structure of a web-portal on the base of Markov decision making process

An approach to optimizing the structure of hyperlinks on a web-portal on the base of Markov decision making processes is regarded. The model describing links between nodes of navigation graph and domain ontology is suggested. Possible ways for applying this approach for e-learning systems are discussed.

Key words: navigation graph, reinforcement learning, Markov decision making process, e-learning.

¹National Kiev-Mohyla Academy University, 04070, Skovorody St.,2, Kyiv, Ukraine
e-mail: oletsky@ukr.net

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

Вступ

Для веб-ресурсів, які характеризуються значною кількістю інформаційних матеріалів та високим ступенем тематичної однорідності та інформаційної зв'язності, очевидним чином виникає проблема оптимізації структури. Йдеться перш за все про оптимізацію гіпертекстових посилань між сторінками веб-сайту так, щоб структура порталу стала більш дружньою для відвідувача, а процес навігації – більш цілеспрямованим. По суті, такий портал має набути експертно-консультаційних рис та надавати консультації відвідувачеві з приводу найбільш доцільних варіантів подальших переходів на інші сторінки.

Поширеним є підхід до пошуку споріднених сторінок, який ґрунтується на аналізі мір близькості між ними [1, 2]. Значною проблемою при цьому є побудова адекватних мір близькості. При цьому слід враховувати те, що найбільш

перспективні переходи пов'язані зі сторінками, схожими на поточну, але не максимально схожими. Крім того, міра перспективності посилань пов'язана не тільки з самими мірами близькості, але і з тим, які сторінки уже були відвідані раніше.

Як альтернативу можна розглядати підхід, який залучає до розгляду аналіз поведінки відвідувачів. Мова може йти як про дослідження історії відвідувань реального порталу, так про агентно-орієнтоване імітаційне моделювання.

Деякі міркування з цього приводу наводилися в [3], але ці міркування носять досить інтуїтивний характер і вимагають подальшої розробки.

Основний зміст роботи

В рамках підходу, що розглядається, поведінку відвідувачів веб-сайту, зокрема тематичного порталу, в загальних рисах можна охарактеризувати як послідовність переходів між

окремими сторінками сайту, спрямовану на досягнення тієї чи іншої мети. При моделюванні процесу досягнення цілей ми розглядаємо два базові підходи:

- драстичний (все або нічого), при якому відвідувач цікавить лише конкретна інформація та успіх досягається лише на сторінці, яка містить цю інформацію, а всі інші розглядаються лише як проміжні ланки на шляху до неї;

- кумулятивний, при якому кожна сторінка може мати самостійну інформаційну цінність, і сумарний інформаційний виграш від процесу навігації певним чином накопичується.

При цьому слід враховувати певні ресурсні обмеження: обмежений загальний час, обмежена кількість переходів, обмежений час перебування на кожній сторінці тощо.

Кожне гіпертекстове посилання можна розглядати як підказку з приводу того, яке рішення на цьому етапі може прийняти відвідувач. Як базову модель, що описує гіпертекстові посилання, природно розглядати структуру, яку можна назвати навігаційним графом. Вузли такого графа відповідають окремим сторінкам, дуги – зв'язкам між ними. Точніше, від вузла А до вузла В йде орієнтована дуга, якщо А містить гіпертекстове посилання на В.

Якщо говорити про проблему оптимізації навігаційного графа, природним чином виникає питання про формальні критерії оптимізації. Ці критерії, безумовно, залежать, від змістовної постановки задачі. На інтуїтивному рівні можна розглядати такі критерії:

- автоматичний підбір посилань, пов'язаних з найбільш імовірними переходами;

- мінімізація кількості переходів, які потрібно здійснити для досягнення цілі;

- досягнення максимального сумарного виграшу при заданих ресурсних обмеженнях.

Звичайно, ці та подібні критерії вимагають більш строгої формалізації.

Моделювання та аналіз процесу навігації має залучати до розгляду моделі, які описують структуру та інформаційне наповнення тематичного порталу, моделі, які характеризують процес досягнення мети відвідувача, а також моделі, які описують власне прийняття рішень. Ключову роль тут відіграє процес прийняття рішень, який можна описати на основі марковських моделей прийняття рішень [4].

Марковський процес прийняття рішень в загальних рисах можна охарактеризувати наступними компонентами:

- множина станів S ;
- множина дій A ;
- функція заохочення R , яка описує винагороду, отриману за дію a в стані s ;
- функція переходу між станами T (взагалі кажучи, вона може мати імовірнісний характер).

При моделюванні процесу навігації стани в найпростішому випадку відповідають окремим сторінкам. Але можна говорити про певні групи станів, які мають спільні характеристики.

Дії відповідають переходам за гіпертекстовими посиланнями. Функція заохочення має відповідати додатковому виграшу, який відвідувач отримав внаслідок переходу.

У найпростішому випадку переходи від стану до стану при кожному конкретному виборі дії можна вважати детермінованими. Але якщо говорити про групи станів, функція переходів може набути імовірнісного характеру.

При аналізі процесу навігації на основі марковських процесів прийняття рішень необхідно отримати деякі рекурентні співвідношення, які дозволяють будувати оцінку корисності поточного стану на основі негайного виграшу від здійснення певної дії та оцінок корисності його наступників [4].

Необхідно звернути увагу на те, що вузли навігаційного графа – це не просто деякі абстрактні вершини. Вони відповідають документам, які, в свою чергу, пов'язані з поняттями предметної області. Тому переходи між вузлами – це фактично переміщення в деякому просторі понять, і врахування цього дозволяє зробити аналіз навігації по сайту більш предметним і семантично-орієнтованим.

У роботах [5, 6] розвивається підхід, спрямований на опис зв'язків між онтологією предметної області та документами, які складають основу інформаційного наповнення порталу з урахуванням специфіки веб-орієнтованих систем. В рамках цього підходу розглядається наступна формалізація зв'язків "онтологія - документ".

Нехай W – множина вузлів онтології предметної області. Будемо вважати, що $W = T \cup V$, де T – множина термінальних концептів (змістовно – множина термінів, які можуть зустрічатися в документах), V – множина метасимволів, тобто нетермінальних концептів, понять, які не можуть безпосередньо з'являтися в документах, але які впливають на їх формування та ранжування за релевантністю.

Можна вважати, що $T \cap V = \emptyset$, оскільки у випадку, якщо слово з тексту водночас є важливим поняттям онтології, ми можемо просто формально перейменувати відповідний нетермінальний концепт.

Аналогічно, множину документів D подамо у вигляді $D = DT \cup DV$, де DT – множина документів як таких, DV – множина категорій документів.

Введемо наступні множини можливих зв'язків:

R – множина типів зв'язків між концептами предметної області (в першу чергу – нетермінальними);

L – множина типів зв'язків між концептами предметної області та документами;

H – множина типів зв'язків між документами.

Тоді модель інформаційного наповнення онтологічно-орієнтованого тематичного порталу можна записати у вигляді відношення

$$M = W^* \cup Z^* \cup D^*, \quad (1)$$

де W^* – множина кортежів (w_1, r, w_2) ; $w_1, w_2 \in W$, $r \in R$;

Z^* – множина кортежів (w, l, d) ; $w \in W$, $l \in L$; $d \in D$;

D^* – множина кортежів (d_1, h, w_2) ; $d_1, d_2 \in D$, $h \in H$.

Співвідношення (1) можна прийняти як деяке робоче наближення до формалізованої моделі. Але воно ще залишається досить інтуїтивним: зокрема, більш точно було б говорити не про те, що зв'язок належить до множини типів зв'язків, а про те, що цей зв'язок є екземпляром деякого класу з множини можливих класів зв'язків. На цьому етапі ця відмінність не грає вирішальної ролі, але подальша формалізація цієї моделі має стати предметом більш ґрунтовного розгляду.

Серед можливих напрямків практичного застосування підходу, що описується, можна відмітити наступні.

1. Оптимізація структури навігаційного графа. Можна сказати, що структура навігаційного графа повинна визначатися закономірностями, які впливають з онтологічного аналізу предметної області. Можна сказати, що структура навігаційного графа тематичного порталу повинна визначатися закономірностями, які впливають з онтологічного аналізу предметної області. Якщо піти далі, можна ставити задачу автоматизованої побудови навігаційного графу на основі аналізу онтології предметної області. Теоретично можна уявляти собі повнозв'язний навігаційний граф, кожна сторінка якого зв'язана з усіма іншими,

але на практиці таку структуру реалізувати неможливо і недоцільно. Користувач бажає бачити не всі можливі переходи, а лише найбільш важливі для нього. Тому постає задача динамічного формування навігаційних графів. В рамках цього підходу слід розрізнити потенційний навігаційний граф, який безпосередньо формується на основі онтології та фактичний навігаційний граф, який утворюється на основі потенційного на основі певних критеріїв оптимізації. Тут процес досягнення цілі більше тяжіє до драстичної функції – відвідувач прагне якомога скоріше перейти до потрібної йому сторінки.

2. Як зворотну задачу можна розглядати побудову онтології певної предметної області на основі аналізу відповідного навігаційного графа. Ці задачі можна розв'язувати незалежно одну від іншої або паралельно.

3. Автоматизований підбір контекстної реклами з найкращими показниками CTR (ефективність власне посилання; ймовірність переходу за посиланням) та CTV (умовна релевантність цільової сторінки; ймовірність здійснення тієї чи іншої дії).

4. Використання в навчальному процесі. Тут вузли навігаційного графа пов'язані з окремими навчальними матеріалами. Можна ввести достатньо адекватний критерій досягнення цілі: підвищення проценту правильних відповідей на тестові запитання. Відповідно до цього процес досягнення цілі описується кумулятивною моделлю. Тоді можна сформулювати проблему адаптації навчального процесу за рахунок автоматичного підбору посилань, які б вели до найбільш корисних ресурсів.

Важливим елементом адаптації навчального процесу має стати кластеризація, тобто розбиття студентів на групи зі схожими характеристиками. Тоді для кожної групи можна отримати свої оцінки корисності станів, і давати рекомендації студентові на основі його віднесення до тієї чи іншої групи.

5. Аналіз поведінки відвідувачів веб-сайту на основі методик Data Mining, зокрема Web Usage Mining [7]. Одну з найбільш типових задач Web Usage Mining можна в найбільш загальних рисах охарактеризувати наступним чином: знаючи історію навігації даного відвідувача, тобто послідовність сторінок $P_1, \dots, P_n$, що були переглянуті цим відвідувачем, виявити закономірності здійснених переходів, і на основі цього – вірогідність того, що він перейде за деяким посиланням на сторінку q , а також

оцінити міру його зацікавленості в цій сторінці. Серед найбільш перспективних напрямків тут можна відмітити:

- побудова дерев рішень з метою формування наборів правил “якщо A , то B ”;
- пошук асоціацій і алгоритм Аргіогі з метою виявлення закономірностей типу “ A і B часто зустрічаються разом”;
- кластерний аналіз з метою розбиття профілів відвідувачів та історій їх навігацій на кластери

Процес навігації на тематичному порталі очевидним чином пов’язаний з процесом прийняття рішень, оскільки кожний перехід – це по суті і є рішення про те, відвідання якої сторінки видається йому найбільш доцільним. Відповідні моделі можна будувати в рамках теорії навчання з підкріпленням на основі марковських процесів прийняття рішень [4]. Дана стаття присвячена розгляду можливих шляхів до застосування цього підходу.

Висновки

В статті розглядається задача оптимізації структури тематичного веб-порталу на основі автоматичного добору гіпертекстових посилань на сторінки, переходи на які з даної сторінки найбільшою мірою сприяють досягненню цілей відвідувачів. Пропонується підхід до розв’язку цієї задачі, пов’язаний з аналізом або моделюванням поведінки відвідувачів на основі марковських процесів прийняття рішень. Підкреслюється, що навігація по порталі - це не просто переходи між абстрактними вузлами, а переміщення по деякій мережі документів, кожний з яких пов’язаний з певними поняттями

предметної області. Розглядається формалізація, спрямована на перехід до аналізу в просторі понять. Побудова більш інтегрованих методик, а також більш строгих формалізацій є предметом подальшого дослідження.

Список використаних джерел

1. Manning C.D., Raghavan P., Schuetze H. Introduction to information retrieval. – Cambridge University Press. – 2008..
2. Lande D. Knowledge search in Internet – M., Publishing House Williams, 2005. (in Russian)
3. Oletsky O. On the problem of simulating visitors flow on the ontologic-oriented subject portal.
4. //Simulation and information technologies. - K.,2010. (in Ukrainian)
5. Russel S., Norvig P. Artificial Intelligence. A modern approach. – Prentice Hall, 2003.
6. Oletsky O. Organization of ontology-oriented means for selecting information on a subject portal. // Scientific notes of NaUKMA. V.99. Computer Science – K., 2009 (in Ukrainian)
7. Oletsky O. Use of graph “ontology-document” for analyzing of visitors behavior on a subject portal.pecypciv. // Scientific notes of NaUKMA. V.125. Computer Science – K., 2011. (in Ukrainian)
8. Barsegyan A., Kupriyanov M., Stepanenko V., Kholod I/. Technologies of data analysis. анализа данных: Data Mining, Visual Mining, Text Mining, OLAP. – St.Peterburg, 2007. (in Russian).

Надійшла до редколегії 29.07.13

УДК 378.016

Омельчук Л.Л., к.ф.-м.н.

**Порівняльний аналіз українського
стандарту
освітньо-професійної підготовки з
інформатики та міжнародних освітніх
стандартів¹**

Наведено порівняльний аналіз українського стандарту освітньо-професійної підготовки з інформатики з міжнародним стандартом Computer Science'2013 та з державним освітнім стандартом Російської Федерації за напрямом підготовки 230100 "Інформатика та обчислювальна техніка". Наведено пропозиції для покращення українського освітнього стандарту.

Ключові слова: інформатика, освітній стандарт.

¹ Частково підтримано Tempus Project INARM

*E-mail: l.omelchuk@gmail.com

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

Вступ

У вересні 2010 року Міністерством освіти і науки України затверджено стандарт освітньо-професійної програми підготовки бакалаврів за напрямом підготовки 040302 "Інформатика" на здобуття кваліфікації 3121 "фахівець з інформаційних технологій" (надалі ОПП'2010) [1].

Для визнання українського диплома з інформатики, а також підвищення мобільності студентів, потрібно щоб український стандарт узгоджувався із стандартами інших країн.

Згідно з дослідженнями [2] основними географічними напрямками міжнародної співпраці тридцяти університетів України є Німеччина, Польща та Росія. Також, в [2, 3] зазначено, що основними країнами-реципієнтами для студентів та молодих учених, що бажають здобути вищу освіту наукові ступені, проходити стажування, навчальну та виробничу практики є Німеччина, США, Росія, Польща та Угорщина. Згідно з цими дослідженнями можна визначити основні напрямки для порівняння українського освітнього стандарту. А саме, в [4] проведемо

L.L. Omelchuk, Ph.D.

**A comparative analysis of Ukrainian
educational standard
in Informatics and international educational
standards²**

A comparative analysis of Ukrainian educational standard in Informatics with international standard Computer Science'2013 and with educational standard of the Russian Federation in the direction of 230100 "Computer Science and Engineering" is made. Proposals for further enhancement of Ukrainian standards are formulated

Key Words: informatics, educational standard.

² Partially supported by Tempus Project INARM

порівняльний аналіз українського стандарту освітньо-професійної програми підготовки бакалаврів з «Інформатики» з широко визнаним у світі міжнародним стандартом Computer Science'2013 (надалі CS'2013) [5], а в [6] – з державним освітнім стандартом Російської Федерації за напрямом підготовки 230100 "Інформатика та обчислювальна техніка" (надалі IBT'2009) [7].

Порівняльна характеристика освітніх стандартів

Автором встановлено відповідність між базовими темами з інформатики та дисциплінами, які пропонуються ОПП'2010, CS'2013 та IBT'2009. Це співвідношення наведено нижче в табл. 1.

Співставляючи дисципліни, з ОПП'2010 з дисциплінами з CS'2013 та IBT'2009 можна побачити, що в CS'2013 є три вибіркові курси, що не знайшли свого відображення в ОПП'2010: "Робототехніка" (IS), "Мобільні платформи" (PDB) та "Цифровий криміналістичний аналіз" (IAS).

Базовий курс CS'2013 "Керування перериваннями та реактивне програмування" (PL), що входить до ядра 2 та на який виділено мінімально 2 години лекційних занять, не представлений в ОПП'2010. Проте, курс входить до ядра 2, деякі дисципліни з якого можуть не читатися. Таким чином, відсутність цього курсу бажано компенсувати, але це не представляє суттєвої проблеми для відповідності програми ОПП'2010 стандарту CS'2013.

Слід звернути особливу увагу на дисципліни, які в CS'2013 чи в ІВТ'2009 є обов'язковими, в той час як в ОПП'2010 вони представлені вибірковими дисциплінами. Так обов'язкова дисципліна "Функціональне програмування" (PL) представлена в ОПП'2010 дисципліною за вибором "Функціональне програмування" (СВ.2.18), дисципліна "Професійна та етична відповідальність" (SP) – двома вибірковими дисциплінами з ОПП'2010: "Етика та естетика" (СВ.5.05) та "Соціальні та професійні питання інформатики" (СВ.3.03). Дисципліна "Інтелектуальна власність" (SP) – дисципліною «Основи авторського права» (СВ.5.13). Дисципліни "Конфіденційність та громадянські свободи" (SP), "Розвиток" (SP) та "Професійна комунікація" (SP) в ОПП'2010 представлені однією дисципліною за вибором "Соціальні та професійні питання інформатики" (СВ.3.03). Обов'язкова дисципліна "Навчання машини" (IS) в ОПП'2010 представлена дисципліною за вибором "Нейронні мережі" (СВ.1.15). А обов'язкова дисципліна "Фундаментальні основи моделювання" (CN) в ОПП'2010 представлена двома вибірковими дисциплінами: "Моделювання складних систем" (СВ.1.12) та "Моделювання інформаційних процесів" (СВ.1.10). Дисципліна ІВТ'2009 "Екологія" (Б.2) в ОПП'2010 представлена дисципліною за вибором "Екологія" (СВ.5.02), дисципліна ІВТ'2009 "Метрологія, стандартизація і сертифікація" (Б.3) в ОПП'2010 представлена дисципліною за вибором "Стандартизація та сертифікація програмного забезпечення ЕОМ" (СВ.2.15), дисципліна ІВТ'2009 "Економіка" (Б.1) в ОПП'2010 представлена дисциплінами за вибором "Економічна теорія" (СВ.5.04) та "Макро- та мікроекономіка" (СВ.5.08).

З іншого боку, є низка дисциплін в ОПП'2010, що не мають свого аналогу в

CS'2013. З обов'язкових дисциплін – це переважно дисципліни гуманітарної та соціально-економічної підготовки (ГСЕ та одна дисципліна професійної та практичної підготовки (ПП): "Українська мова (за професійним спрямуванням)" (ГСЕ 1.01), "Історія української культури" (ГСЕ 1.03), "Іноземна мова" (ГСЕ 1.04), "Філософія" (ГСЕ 1.05), "Фізичне виховання (за рахунок вільного часу студента)" (ГСЕ 1.06). Не знайшли свого аналогу в ІВТ'2009: "Історія української культури" (ГСЕ 1.03), "Інтелектуальні інформаційні системи" (ПП 3.06), "Методи оптимізації та дослідження операцій" (ПП 3.08), "Методика викладання математики та інформатики" (ПП 3.09), "Організація та обробка електронної інформації" (ПП 3.12), "Розподілені інформаційно-аналітичні системи" (ПП 3.18), "Системний аналіз та теорія прийняття рішень" (ПП 3.19), "Теорія програмування" (ПП 3.20).

Крім того ціла низка вибіркових дисциплін в ОПП'2010 не знайшли свого аналогу в CS'2013 та в ІВТ'2009: "Методи стохастичної оптимізації" (СВ.1.05), "Мови обчислювань та кластерні системи" (СВ.1.06), "Розпізнавання образів" (СВ.1.19), "Символьні обчислення та комп'ютерна алгебра" (СВ.1.20), "Теорія ігор" (СВ.1.22), "Теорія керування" (СВ.1.24), "Теорія масового обслуговування" (СВ.1.26), "Фізика" (СВ.1.27), "Функціональний аналіз" (СВ.1.28), "Unix-подібні операційні системи" (СВ.2.01), "Технологія XML" (СВ.2.04), "Комп'ютерна вірусологія" (СВ.2.06), "Стандартизація та сертифікація програмного забезпечення ЕОМ" (СВ.2.12), "Інформаційні системи підприємств" (СВ.3.01), "Менеджмент знань" (СВ.3.02), "Організаційна інформатика" (СВ.3.03), "Основи педагогіки" (СВ.4.01), "Бухоблік та аудит" (СВ.5.01), "Екологія" (СВ.5.02), "Економетрія" (СВ.5.03), "Зарубіжна філософія" (СВ.5.06), "Іноземна мова (за професійним спрямуванням – інформатика)" (СВ.5.07), "Макро- та мікроекономіка" (СВ.5.08), "Маркетинг" (СВ.5.09), "Менеджмент" (СВ.5.10), "Методи наукових досліджень" (СВ.5.12), "Політологія" (СВ.5.15), "Правознавство" (СВ.5.16), "Релігієзнавство" (СВ.5.18), "Соціологія" (СВ.5.19), "Філософія природознавства" (СВ.5.20).

Таблиця 1. Порівняння CS'2013, ООП'2013, ІВТ'2009

CS'2013		ООП'2013		ІВТ'2009	
Перелік дисциплін	Обов'язковість [унікальність до ООП'2013]	Перелік дисциплін	Обов'язковість [унікальність до CS'2013/унікальність до ІВТ'2009]	Перелік дисциплін	Обов'язковість [унікальність до ООП'2013]
(1)	(2)	(3)	(4)	(5)	(6)
“Функціональне програмування” (“Functional Programming”) (PL)	+/+	“Функціональне програмування” (СВ.2.18)	-/+/-	-	
“Професійна та етична відповідальність” (“Professional Ethics”) (SP)	+/+	“Етика та естетика” (СВ.5.05)	-/-/-	-	
		“Соціальні та професійні питання інформатики» (СВ.3.03)	-/-/-	-	
“Інтелектуальна власність” (“Intellectual Property”) (SP)	+/+	«Основи авторського права» (СВ.5.13)	-/+/-	-	
“Конфіденційність та громадянські свободи” (“Privacy and Civil Liberties”) (SP)	+/-	“Соціальні та професійні питання інформатики” (СВ.3.03)	-/+/-	-	
“Розвиток” (“Sustainability”) (SP)	+/-				
“Професійна комунікація” (“Professional Communication”) (SP)	+/-				
“Навчання машини” (“Basic Machine Learning”) (IS)	+/+	“Нейронні мережі” (СВ.1.15)	-/+/-	-	

Продовження табл. 1

(1)	(2)	(3)	(4)	(5)	(6)
“Фундаментальні основи моделювання” (“Fundamentals”) (CN)	++	“Моделювання складних систем” (CB.1.12)	-/-/-	-	
		“Моделювання інформаційних процесів” (CB.1.10)	-/-/-	-	
“Робототехніка” (“Robotics”) (IS),	-	-		-	
“Мобільні платформи” (“Mobile Platforms”) (PDB)	-	-		-	
“Цифровий криміналістичний аналіз” (“Digital Forensics”) (IAS)	-	-		-	
-		“Українська мова (за професійним спрямуванням)” (ГСЕ 1.01)	-/+/-	-	
-		“Іноземна мова” (ГСЕ 1.04)	+ /+/-	“Іноземна мова” (“Иностранный язык”) (Б.1)	++
-		“Філософія” (ГСЕ 1.05)	+ /+/-	“Філософія” (“Философия”) (Б.1)	++
-		“Фізичне виховання (за рахунок вільного часу студента)” (ГСЕ 1.06)	+ /+/-	“Фізичне виховання” (“Физическая культура”) (Б.1)	++
-		“Методика викладання математики та інформатики” (ПП 3.09)	+	-	
-		“Історія української культури” (ГСЕ 1.03),	+	-	

Продовження табл. 1

(1)	(2)	(3)	(4)	(5)	(6)
“Основні питання, пов’язані з інтелектуальними системами” (“Fundamental Issues”) (IS)	+/-	“Інтелектуальні інформаційні системи” (ПП 3.06)	+/-/-	-	
Базові стратегії пошуку (Basic Search Strategies) (IS)	+/-				
“Представлення знань та виведення” (“Basic Knowledge Representation and Reasoning”) (IS)	+/-				
“Поглиблене вивчення пошуку” (“Advanced Search”) (IS)	-/-				
“Поглиблене вивчення представлення знань та виведення” (“Advanced Representation and Reasoning”) (IS)	-/-				
“Розподіл ресурсів” (“Resource Allocation”) (NC)	+/-	“Методи оптимізації та дослідження операцій” (ПП 3.08)	+/-/-	-	
“Обробка” (“Processing”) (CN)	-/-				
“Розподіл ресурсів та планування” (“Resource Allocation and Scheduling”) (SF)	+/-				

Продовження табл. 1

(1)	(2)	(3)	(4)	(5)	(6)
“Співробітництво та комунікація” (“Collaboration & communication”) (HC)	-/-	“Організація та обробка електронної інформації” (ПП 3.12)	+/-/-	-	
“Управління інформацією” (“Information Management Concepts”) (IM)	+/-				
“Моделювання даних” (“Data Modeling”) (IM)	+/-				
“Надійність через надлишковість” (“Reliability through Redundancy”) (SF)	+/-				
“Дані, інформація та знання” (“Data, Information, and Knowledge”) (CN)	-/-				
“Керування ризиками” (“Risk Management”) (IAS)	-/-	“Розподілені інформаційно-аналітичні системи” (ПП 3.18)	+/-/-	-	
“Управління інформацією” (“Information Management Concepts”) (IM)	+/-				

Продовження табл. 1

(1)	(2)	(3)	(4)	(5)	(6)
“Фундаментальні поняття програмування” (“Fundamental Programming Concepts”) (SDF)	+/-	“Системний аналіз та теорія прийняття рішень” (ПП 3.19),	+ / + / -	-	
“Методи розвитку” (“Development Methods”) (SDF)	+/-				
“Обробка природної мови” (“Natural Language Processing”) (IS)	-/-	“Теорія програмування” (ПП 3.20)	+ / + / -	-	
“Верифікація і валідація ПЗ” (“Software Verification Validation”) (SE)	+/-				
“Формальні методи” (“Formal Methods”) (SE)	-/-				
“Основи обчислень” (“Basics of Counting”) (DS)	+/-	“Алгебра та геометрія” (МПН 2.01)	+ / + / -	“Математика” (“Математика”) (Б.2)	+ / +
“Функції, відношення та множини” (“Sets, Relations, and Functions”) (DS)	+ / +	“Дискретна математика” (МПН 2.02)	+ / + / -		
“Графи та дерева” (“Graphs and Trees”) (DS)	+ / +				

Продовження табл. 1

(1)	(2)	(3)	(4)	(5)	(6)
“Основи обчислень” (“Basics of Counting”) (DS)	+/-	“Диференціальні рівняння” (МПН 2.03)	+ /+/-		
“Транзитивний автомат” (“State-State Transition-State Machines”) (SF)	+/+	“Математична логіка та теорія алгоритмів” (МПН 2.04)	+ /+/-		
“Основи аналізу алгоритмів” (“Basic Analysis”) (AL)	+/-				
“Основи теорії автоматів та обчислювальності складності” (“Basic Automata, Computability and Complexity”) (AL)	+/+				
“Поглиблене вивчення теорії автоматів та обчислюваності” (“Advanced Automata Theory and Computability”) (AL)	-/+				
“Основи логіки” (“Basic Logic”) (DS)	+/+				
“Методи доведення” (“Proof Techniques”) (DS)	+/+				

Продовження табл. 1

(1)	(2)	(3)	(4)	(5)	(6)
“Основи обчислень” (“Basics of Counting”) (DS)	+/-	“Математичний аналіз” (МПН 2.05)	+ /+/-		
“Дискретна ймовірність” (“Basics of Counting”) (DS)	++	“Теорія ймовірностей та математична статистика” (МПН 2.06)	+ /+/-		
-		“Екологія” (СВ.5.02)	-/-/+	“Екологія” (“Экология”) (Б.2)	++
-		“Стандартизація та сертифікація програмного забезпечення ЕОМ” (СВ.2.15)	-/-/+	“Метрологія, стандартизація та сертифікація” (“Метрология, стандартизация и сертификация”) (Б.3),	++
		“Економічна теорія” (СВ.5.04)	-/-/-	“Економіка” (“Экономика”) (Б.1)	++
-		“Макро- та мікроекономіка” (СВ.5.08)	-/-/-		
“Відмовостійкість” (“Fault Tolerance”) (OS)	-/+	“Архітектура обчислювальних систем” (ПП 3.02).	+ /+ /+	“Електротехніка, електроніка і схемотехніка” (“Электротехника, электроника и схемотехника”) (Б.3)	+/-
“Оцінка продуктивності системи” (“System Performance Evaluation”) (OS)	-/+			“ЕОМ і периферійні пристрої” (“ЭВМ и периферийные устройства”) (Б.3)	+/-
“Керування пам'яттю” (“Memory Management”) (OS)	++				
“Віртуальні машини” (“Virtual Machines”) (OS)	-/+				

Продовження табл. 1

(1)	(2)	(3)	(4)	(5)	(6)
-		“Фізика” (СВ.1.27)	-/-/+	“Фізика” (“Физика”) (Б.2)	+/+
-		“Іноземна мова (за професійним спрямування м інформатика) ” (СВ.5.07)	-/-/+	“Іноземна мова” (“Иностранны й язык”) (Б1)	+/+
-		“Методи стохастичної оптимізації” (СВ.1.05)	-	-	
-		“Мови обчислювань та кластерні системи” (СВ.1.06)	-	-	
-		“Розпізнаван ня образів” (СВ.1.19)	-	-	
-		“Символьні обчислення та комп’ютерна алгебра” (СВ.1.20)	-	-	
-		“Теорія ігор” (СВ.1.22)	-	-	
-		“Теорія керування” (СВ.1.24),	-	-	
-		“Теорія масового обслуговуван ня” (СВ.1.26)	-	-	
-		“Функціональ ний аналіз” (СВ.1.28)	-	-	
-		“Unix-подібні операційні системи” (СВ.2.01)	-	-	
-		“Технологія XML” (СВ.2.04)	-	-	
-		“Комп’ютерн а вірусологія” (СВ.2.06)	-	-	
-		“Інформаційн і системи підприємств” (СВ.3.01)	-	-	

Продовження табл. 1

(1)	(2)	(3)	(4)	(5)	(6)
-		“Менеджмент знань” (СВ.3.02)	-	-	
-		“Організацій на інформатика” (СВ.3.03),	-	-	
-		“Основи педагогіки” (СВ.4.01)	-	-	
-		“Бухоблік та аудит” (СВ.5.01)	-	-	
-		“Економетрія ” (СВ.5.03)	-	-	
-		“Зарубіжна філософія” (СВ.5.06)	-	-	
-		“Маркетинг” (СВ.5.09)	-	-	
-		“Менеджмент ” (СВ.5.10)	-	-	
-		“Методи наукових досліджень” (СВ.5.12)	-	-	
-		“Політологія” (СВ.5.15)	-	-	
-		“Правознавст во” (СВ.5.16)	-	-	
-		“Релігієзнавс тво” (СВ.5.18)	-	-	
-		“Соціологія” (СВ.5.19)	-	-	
-		“Філософія природознавс тва” (СВ.5.20)	-	-	

Висновки

З проведеного дослідження видно, що для забезпечення відповідності українського освітнього стандарту рекомендаціям CS'2013 та ІВТ'2009 слід читати наступні дисципліни за вибором: "Нейронні мережі" (СВ.1.15), "Функціональне програмування" (СВ.2.18), "Основи авторського права" (СВ.5.13), "Соціальні та професійні питання інформатики" (СВ.3.03), принаймні одна із дисциплін: "Моделювання складних систем" (СВ.1.12), "Моделювання інформаційних процесів" (СВ.1.10) та принаймні одна із дисциплін: "Етика та естетика" (СВ.5.05), "Соціальні та професійні питання інформатики" (СВ.3.03). А також, "Екологія" (СВ.5.02), "Стандартизація та сертифікація програмного забезпечення ЕОМ" (СВ.2.15) та принаймні одна з двох дисциплін "Економічна теорія" (СВ.5.04) чи "Макро- та мікроекономіка" (СВ.5.08). Крім того, бажаним є введення до ОПП'2010 курсу, який би відповідав базовому курсу CS'2013 "Керування перериваннями та реактивне програмування" (PL).

Введення цих змін полегшить українським університетам співпрацю з закордонними університетами, а також спростить українським студентам можливість отримання вищої освіти наукових ступенів, проходження стажування,

навчальної та виробничої практики в країнах Європи, США та Російській Федерації.

Список скорочень:

У даній статті застосовуються наступні скорочення:

ПЗ – програмне забезпечення;
ОПП'2010 – галузевий стандарт вищої освіти підготовки бакалаврів за напрямом 040302 «Інформатика» 2010;
CS'2013 – Computer Science Curricula 2013;
ГСЕ – гуманітарної та соціально-економічної підготовки;
МПН – математичної, природничо-наукової підготовки;
ПП – професійної та практичної підготовки;
СВ – самостійного вибору
ІВТ'2009 – Федеральный государственный образовательный стандарт высшего профессионального образования по направлению подготовки 230100 "Информатика и вычислительная техника";
НЦ ООП – навчальний цикл основної освітньої програми;
Б – базової підготовки.

Список використаних джерел

1. The industry standard of the higher education of preparation of bachelors behind the direction 040302 «Informatics». Kyiv, 2010. (in Ukrainian).
2. Shava A. Migration processes in education - development dynamics, causes and current state [Electronic resource]. - Mode of access: http://www.ippo.if.ua/images/stories/Obr_2012/Obrii_1_12/Shava.pdf (in Ukrainian).
3. Kutsenko V. Education Migration: Socio-economic and professional aspects / V. Kutsenko N. Rudenko // Regional Economy. - 2010. - № 1. - P. 123-130. (in Ukrainian).
4. Omelchuk L.L. Nikitchenko M.S. Comparative analysis of the Ukrainian standard program for professional training in science and international standard Computer Science'2013 // Actual problems of ICT training (APICTST-2013). (in Ukrainian).
5. Computer Science Curriculum 2013. Strawman Draft. The Joint Task Force on Computing Curricula, Association for Computing Machinery, IEEE-Computer Society.
6. Omelchuk LL Comparative analysis of the Ukrainian standard educational training in computer science in 2010 and the state educational standard of higher professional education of Computer Science of the Russian Federation in 2009// Actual problems of ICT training (APICTST-2013). (in Ukrainian).
7. Federal state educational standard of higher education in the field of training 230100 «Computer Science». Moscow, 2009. (in Russian).

Надійшла до редколегії 10.09.2013

УДК 81'322, 004.8

Порхун О.В., к.ф.-м.н., асист.

Огляд базових моделей для вирішення задачі кореференс

Дана стаття присвячена огляду підходів для вирішення задачі кореференс. Розглядаються моделі побудови векторів ознак для згадувань у текстах: модель «пара згадувань», модель «кластер-згадування», модель «ранжування згадувань» та модель «ранжування кластерів». А також описано застосування машини опорних векторів у відповідності з моделлю побудови векторів ознак згадувань для вирішення задачі кореференс.

Ключові слова: відношення кореферентності, кластер, ранжування, машина опорних векторів (SVM).

Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Глушкова 4д, e-mail: elena_porkhun@mail.ru

Статтю представив чл.-кор. НАН України, д.ф.-м.н., проф. Анісімов А.В.

Відношення кореферентності – це відношення між іменами – компонентами висловлення, в якому імена посилаються на один і той самий об'єкт (ситуацію) зовні мовної дійсності (референт). З використанням відношення кореферентності текст можна зробити змістовно зв'язним, часто, але необов'язково, у вигляді анафоричного відношення. Наприклад, анафоричним відношенням займенникове слово - займенник може бути: «Книгою користується читач. Вона цікава». Прикладом анафоричного відношення значення невизначеності - займенник є: «Подарунок здивує іменника. Подорож незвичайна».

Більшість досліджень та публікацій в області вирішення задач кореференс базуються на розробці методів для автоматичного встановлення кореферентних зв'язків у природно-мовному тексті. У даній статті пропонується огляд моделей для побудови векторів ознак згадувань, які перевіряються на відношення кореферентності.

Як приклад будемо розглядати фрагмент тексту, наведеного на Рис.1.

O.V. Porkhun, PhD.

Review of the basic models for coreference resolution problem

This article deals with review of approaches for coreference resolution problem. The models of constructing feature vectors for mentions in texts: Mention-Pair Model, Entity-Mention Model, Mention-Ranking Model and Cluster-Ranking Model are considered. Also application of support vector machine in accordance with the model of constructing feature vectors of mentions for coreference resolution problem is described.

Key Words: coreference relation, cluster, ranking, support vector machine (SVM).

Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova st., 4d, e-mail: elena_porkhun@mail.ru

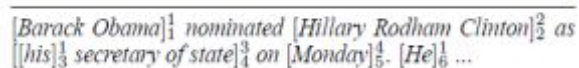


Рис.1. Приклад розміченого фрагменту тексту

Кожне згадування m в даному фрагменті позначається як $[m]_{mid}^{cid}$, де mid – це номер згадування та cid – номер кластера, до якого m належить. У прикладі згадування поділяються на 4 кластери: *Barack Obama*, *his*, та *he* в одному кластері; *Hillary Rodham Clinton*, *secretary of state* та *Monday* – інші кластери.

Виділяють 4 моделі для побудови векторів ознак згадувань, що перевіряються на відношення кореферентності: «пара згадувань», «кластер-згадування», «ранжування згадувань» та «ранжування кластерів» [1].

Модель «пара згадувань»

Дана модель є класифікатором, що вирішує чи є кореферентною пара згадувань – поточного активного згадування m_k та попереднього кандидата згадування m_j . Кожний приклад $i(m_j, m_k)$ представляє згадування m_j та m_k та задається множиною виділених ознак. Відповідь

класифікатора – 1 або 0, залежно від того чи є кореферентними m_j та m_k .

За даною моделлю формується позитивний приклад для згадування m_k та його найближчого антицидента m_j та негативний приклад для m_k , що співставляється з кожним проміжним згадуванням $m_{j+1}, m_{j+2}, \dots, m_{k-1}$. Для прикладу з рис.1 буде згенеровано 3 навчальні приклади для $He: i(\text{Monday}, He), i(\text{secretary of state}, He)$ та $i(\text{his}, He)$. Перші два будуть позначені як негативні, останній – як позитивний.

Після навчання результат класифікатора використовується для ідентифікації антициденту для поточного згадування у тестовому тексті. А саме, поточне згадування m_k , що розглядається, вибирає в якості антицидента найближче згадування, що стоїть перед ним, яке перевіряється на відношення кореферентності з ним. Якщо m_k не класифікується як кореферентне з жодним згадуванням, що стоять перед ним, то m_k розглядається як новий дискурс (discourse-new).

Модель «кластер-згадування»

Модель «кластер-згадування» представляє собою класифікатор, який вирішує чи є кореферентними поточне активне згадування m_k та попередньо сформований кластер c_j . Кожен навчальний приклад $i(c_j, m_k)$ включає кластер c_j та згадування m_k .

Ознаки рівня кластера створюються з ознак для пар згадувань використовуючи 4 предикати: NONE, MOST-FALSE, MOST-TRUE та ALL. А саме, для кожної ознаки x спочатку проводиться її перетворення в еквівалентну множину ознак з бінарними значеннями, якщо вона є багатозначною. Потім для кожної бінарної ознаки x_b створюється 4 бінарні ознаки рівня кластера:

(1) NONE- X_b дорівнює true, якщо x_b дорівнює false між m_k та кожним згадуванням з c_j ;

(2) MOST-FALSE- X_b дорівнює true, якщо x_b дорівнює true між m_k та менш ніж половиною (але принаймні одним із) згадувань з c_j ;

(3) MOST-TRUE- X_b дорівнює true, якщо x_b дорівнює true між m_k та не менш ніж половиною (але не з усіма) згадувань з c_j ;

(4) ALL- X_b дорівнює true якщо x_b дорівнює true між m_k та кожним згадуванням з c_j .

Таким чином, згідно моделі «кластер-згадування» формується позитивний приклад для кожного поточного згадування m_k та попередньо сформованого кластера c_j , до якого m_k належить, та негативний приклад для m_k , що співставляється з кожним кластером, чие останнє

згадування з'являється між m_k та найближчим до нього антицидентом (тобто останнє згадування з c_j). Для прикладу з рис.1 буде згенеровано 3 навчальні приклади:

$He: i(\{\text{Monday}\}, He), i(\{\text{secretary of state}\}, He)$, та $i(\{\text{Barack Obama, his}\}, He)$. Перші два будуть позначені як негативні, останній – як позитивний.

Після навчання результат класифікатора використовується для визначення чи є кореферентними кластер та згадування у тестовому тексті. А саме, згадування обробляються зліва направо. Для кожного активного згадування m_k створюється тестовий приклад між ним та кожним із сформованих вже кластерів. Всі тестові приклади предствляються класифікатору. В результаті m_k буде пов'язаний з найближчим сформованим кластером, який виявився кореферентним з m_k . Якщо m_k виявився некорееферентним з жодним із кластерів, він розглядається як новий дискурс. Всі кластери до розгляду згадування m_k формуються послідовно на основі результатів класифікатора для перших $k - 1$ згадувань.

Модель «ранжування згадувань»

Дана модель накладає ранжування на всіх кандидатів антицидентів активного згадування m_k . Аналогічно моделі «пара згадувань» кожен навчальний приклад $i(m_j, m_k)$ містить поточне згадування m_k та згадування m_j , яке йому передуює.

Характерною особливістю моделі є те, що навчальним прикладам присвоюються оцінки, значення класів (class values).

Нехай S_k – множина навчальних прикладів, створена для анафоричного згадування m_k , значення класу для прикладу $i(m_j, m_k)$ в S_k – ранг m_j серед конкуруючих кандидатів антицидентів, який дорівнює 2, якщо m_j – найближчий антицидент m_k та 1 – в іншому випадку.

Для нашого текстового фрагменту з Рис.1 маємо 3 навчальні приклади для $He: i(\text{Monday}, He), i(\text{secretary of state}, He), i(\text{his}, He)$. Третій приклад буде мати ранг 2, решта – ранг 1.

Після навчання модель ранжування згадувань застосовується для ранжування кандидатів антицидентів для активного згадування у тестовому тексті. Для активного згадування m_k застосовується незалежний класифікатор для визначення чи є m_k новим дискурсом. Якщо так, то дане згадування не розглядається далі. В іншому випадку, створюється тестовий приклад для m_k шляхом співставлення його з кожним згадуванням, що стоїть перед ним. Потім тестовий приклад представляється класифікатору для ранжування і згадування з найвищим рангом,

з яким співставляється m_k , вибирається як антицидент m_k .

Модель «ранжування кластерів»

Мета даного підходу – комбінація моделей кластер-згадування та ранжування згадувань.

Даний підхід може мати 2 реалізації.

1. Розглядати класифікатор з ранжуванням як конвеєрну архітектуру, де виявлення нових дискурсів виконується перед задачею кореференс.

2. Побудувати класифікатор з ранжуванням, який одночасно розв'язує задачу виявлення нових дискурсів та задачу кореференс.

У процесі кластерного ранжування визначаються ранги множини кластерів для активного згадування m_k . Представлення прикладів аналогічне як у моделі «кластер-згадування», де кожен навчальний приклад $i(c_j, m_k)$ представляє кластер c_j і згадування m_k та містить ознаки рівня кластера, сформовані із значень предикатів. На відміну від моделі «кластер-згадування» у кластерному ранжуванні навчальний приклад створюється між кожним старим дискурсом згадуванням m_k і кожним з кластерів; присвоєння рангів (класових оцінок) навчальним прикладам реалізується аналогічно моделі «ранжування згадувань». А саме, ранг навчального прикладу $i(c_j, m_k)$ створеного для m_k – це ранг кластера c_j серед конкуруючих кластерів, який дорівнює 2, якщо m_k належить c_j та 1 – в іншому випадку.

Застосування кластерного ранжування до тестового тексту ідентичне застосуванню моделі «ранжування згадувань». Згадування обробляються зліва направо. Для кожного активного згадування m_k застосовується незалежний класифікатор для визначення того, чи є згадування m_k новим дискурсом. Якщо так, то дане згадування m_k не розглядається далі. Інакше, створюється тестовий приклад для m_k шляхом співставлення його з кожним сформованим кластером. Потім тестовий приклад представляється класифікатору для ранжування та m_k відноситься до того кластеру, який отримав найвищий ранг.

Об'єднання задач визначення нових дискурсів та кореференс

Для об'єднання задач визначення нових дискурсів та кореференс, потрібно навчити класифікатор, що використовує приклади, згенеровані як для старих дискурсів, так і для нових дискурсів. А саме, при навчанні класифікатора у моделі «ранжування кластерів» забезпечується можливість для кожного

активного згадування формувати новий кластер. Це забезпечується шляхом введення додаткового прикладу, який (1) включає ознаки, які описують лише дане активне згадування та (2) мають найвищий ранг серед всіх конкуруючих кластерів (тобто 2), якщо дане активне згадування є новим дискурсом та найменший ранг (тобто 1) – в іншому випадку. Головна перевага об'єднуючого підходу полягає в тому, що він дозволяє оцінювати всі можливі опції для активного згадування (чи є згадування старим, якщо так, то який із сформованих кластерів є найкращим для нього) одночасно.

Таким чином, після навчання для кожного активного згадування m_k створюються тестові приклади шляхом співставлення активного згадування з кожним із сформованих кластерів. Для реалізації можливості бути m_k новим дискурсом створюється додатковий тестовий приклад, що містить ознаки, які описують лише дане активне згадування. Якщо при пред'явленні додаткового тестового прикладу класифікатору він отримає найвищий ранг, то дане активне згадування класифікується як новий дискурс і більше не розглядається. Інакше, m_k відноситься до того кластеру, який має найбільший ранг. Всі кластери, що передують m_k формуються послідовно на основі результатів класифікатора для перших $k - 1$ згадувань.

Застосування машини опорних векторів для вирішення задачі кореференс

Одним із ефективних методів машинного навчання, що добре зарекомендував себе при вирішенні задачі кореференс, є метод опорних векторів. Зокрема, результати його застосування наведені в роботах [1,2,3,4].

Основна ідея методу опорних векторів (SVM) – перетворення вхідних векторів у простір більш високої розмірності, де є висока ймовірність, що дані будуть лінійно-роздільними, та пошук роздільної гіперплощини з максимальним зазором у цьому просторі. У випадку вирішення задачі кореференс вхідні вектори формуються у вигляді прикладів відповідно до обраної моделі представлення (пари згадувань, кластер-згадування, ранжування згадувань чи кластерне ранжування) та складаються з виділених ознак. Ознаки, що були використані для вирішення задачі кореференс з використанням описаних вище моделей, наведені в [1].

Дві паралельні гіперплощини будуються по обидві сторони гіперплощини, яка розділяє 2 класи. Роздільною гіперплощиною є гіперплощина, яка максимізує відстань до двох

паралельних гіперплощин. При цьому таку гіперплощину називають *оптимальною гіперплощиною*, а точки даних, які лежать найближче до цієї гіперплощини, називаються *опорними векторами*. Алгоритм працює таким чином, що чим більша відстань буде між паралельними гіперплощинами, тим менша середня помилка класифікатора.

Для формального опису задачі нехай точки, які класифікуються, мають вигляд: $\{(x_1, d_1), (x_2, d_2), \dots, (x_n, d_n)\}$, де d_i приймає значення 1 або -1, в залежності від того, якому класу належить точка x_i . Кожна точка x_i – p -вимірний дійсний вектор. У нашому випадку, x_i – це вектор ознак відповідно до обраної моделі представлення прикладів, а d_i – це значення, що характеризує наявність чи відсутність кореферентності: 0 або 1, або значення рангу, якщо використовується у моделях ранжування згадувань та кластерного ранжування.

Для класифікації точок з даної навчальної вибірки роздільна гіперплощина має вигляд:

$$w \cdot x + b = 0,$$

де вектор w – перпендикуляр до роздільної гіперплощини. Параметр b визначає відстань гіперплощини до початку координат. Якщо параметр b дорівнює нулю, гіперплощина проходить через початок координат.

Класифікація на основі SVM полягає в наступному:

- 1) Вхідні вектори подаються на вхід SVM у вигляді прикладів відповідно до обраної моделі представлення та складаються з виділених ознак.
- 2) Бажані значення d_i – це значення, що характеризують наявність чи відсутність кореферентності: 0 або 1, або значення рангу, якщо використовується у моделях ранжування згадувань та кластерного ранжування.
- 3) Навчання SVM базується на розв'язанні задачі квадратичного програмування з використанням методу множників Лагранжа.
- 4) Для випадку лінійної нероздільності у цільову функцію замість скалярних добутків вводиться нелінійна функція ядра.
- 5) У результаті навчання SVM отримаємо оптимальний вектор вагових коефіцієнтів, що визначає перпендикуляр до роздільної гіперплощини та оптимальне значення порогу, що задає відстань гіперплощини до початку координат. Знайдені параметри підставляються у рівняння гіперплощини для нових точок – пар згадувань у текстах з тестової вибірки чи пар кластер-згадувань відповідно до обраної моделі

побудови векторів ознак, таким чином здійснюється вирішення задачі кореференс.

Список використаних джерел

1. *Altaf Rahman, Vincent Ng. Supervised Models for Coreference Resolution // Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, 6-7 August 2009, Singapore. – P. 968–977.*
2. *Uryupina O. Corry: A System for Coreference Resolution // Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, July 15-16, 2010, Uppsala, Sweden. – P. 100–103.*
3. *Huiwei Zhou, Yao Li, Degen Huang, Yan Zhang, Chunlong Wu, Yuansheng Yang. Combining Syntactic and Semantic Features by SVM for Unrestricted Coreference Resolution // Proceedings of the 15th Conference on Computational Natural Language Learning: Shared Task, June 23-24, 2011, Portland, Oregon. – P. 66–70.*
4. *Finley Th., Joachims Th. Supervised Clustering with Support Vector Machine // Proceedings of the 22nd International Conference on Machine Learning, 2005, Bonn, Germany. – P. 217-224.*

Надійшла до редколегії 07.10.13

УДК 004.652

Рубан Н.Н.¹

Использование графовых баз данных при обработке реляционных баз данных

Рассмотрен метод совместного использования реляционных баз данных и баз данных на основе графов, их преимущества относительно реляционных баз при решении некоторых задач.

Ключевые слова: графовые базы данных, граф, Neo4j, MySQL.

¹ Донбасский государственный педагогический университет, 84122, г. Славянск, ул. Г.Батюка 19.

e-mail: rubannn@gmail.com

Статью представил д.ф.-м.н., проф. Буй Д.Б.

Введение

Достаточно часто при обработке данных, которые хранятся в реляционной базе, возникает потребность в их анализе. Но зачастую, реляционная структура не может удовлетворить все необходимые потребности, либо обработка такой структуры сложна и трудоемка. Поэтому предлагается совместное использование различных по структуре баз данных, а именно связка MySQL [2] (реляционная) и Neo4j [1] (графовая).

Графовая база данных – это база данных, структура которой представлена в виде графа, при этом данные хранятся либо в вершинах графа, либо в их ребрах.

Базы данных на основе графа не ограничиваются predetermined отношениями и ограниченным набором атрибутов. Графовая база данных содержит много узлов, у которых есть свойства, между узлами в графе существует отношение (связь), у которого могут также быть атрибуты. Нет никаких ограничений на то, сколько узлов база данных может иметь, сколько отношений существует между узлами, или сколько свойств данный узел или отношение могут иметь. Это предполагает очень динамическое представление соответствующих данных, которые могут быть сохранены, в виде произвольных узлов которые в свою очередь могут быть связаны произвольным числом отношений, у которых также может быть

N.N. Ruban

Using the graph database in the processing relational database

Describes the method of shared use relational databases, and graph databases, and their advantages of relational databases in solving some tasks.

Key words: graph databases, graph, Neo4j, MySQL

Donbass State Teachers' Training University, 84122, c. Slavyansk, 19 G. Batuka street

любое число свойств. Структура и расположение данных в графовой базе графика являются очень динамичными, и очень гибкими к типам данных, которые они могут хранить.

Таким образом, база данных в виде графа должна соответствовать следующим критериям:

- хранение оптимизировано для данных, представленных в виде графа, с обеспечением хранения данных по вершинам и ребрам;
- хранение оптимизировано для просмотра графа без использования индекса. Графовая база данных оптимизирована для запросов, используя близость данных, начиная от одного или нескольких корневых узлов, а не использование глобальных запросов;
- гибкая модель данных для некоторых решений: нет необходимости в определении типов данных для вершин и ребер в отличие от более ограниченной табличной модели реляционной базы данных;
- встроенная API с точкой входа для классических алгоритмов теории графов (кратчайший путь, алгоритм Дейкстры, A* и др.).

Следует отметить что в Neo4j, используется два различных алгоритма, поиска кратчайшего пути между двумя узлами. Первый включенный алгоритм это алгоритмом Дейкстры, который

внедрен в стандартный класс graphAlgoFactory. Второй алгоритм поиска кратчайшего пути также доступен в graphAlgoFactory, но он базируется на алгоритме обхода графа в ширину, с последующим вычислением кратчайшего пути.

Рассмотрим один из возможных механизмов взаимодействия MySQL и Neo4j. Опишем условия, которые были выбраны для доступа к данным:

1. База данных MySQL является основным хранилищем данных.

2. Neo4j является вторичным хранилищем данных, которое содержит в себе подмножество основных данных и отвечает следующим условиям:

а). поддержка OLTP (Online Transaction Processing), доступ к заранее сформированным наборам данных в реальном времени, которые соответствуют определенным условиям.

б). поддержка пакетного режима – сбор и обработка данных с некоторой задержкой по времени (либо по расписанию).

Таким образом, необходимо иметь быстрый доступ к имеющемуся набору данных, а также возможность синхронизации данных между двумя системами.

Основная часть

Рассмотрим данный подход на примере. Смоделируем базу данных отношений между поставщиками и клиентами в сфере продажи автомобилей. Представим каждый элемент в виде объекта со своими методами (рис. 1).

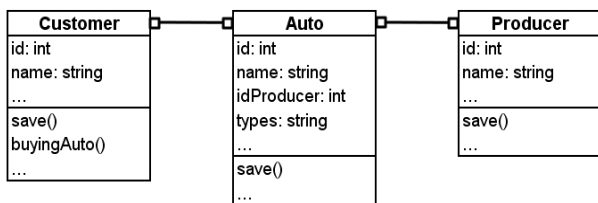


Рис. 1 Модель реляционной базы данных

Объекты связаны посредством идентификатора, который является первичным ключом, индексирование происходит по данному ключу. Методы каждого объекта следят за сохранением данных в структуру Neo4j, преобразовывая их в узлы, а связи – в ребра графа (рис. 2).

При импорте данных в реляционную базу данных важно следить за корректным преобразованием типов и сохранением данных. Для этого можно использовать SQL-запросы, выбирая при этом только необходимые данные,

либо использовать встроенную систему API, которая позволяет импортировать/экспортировать данные.

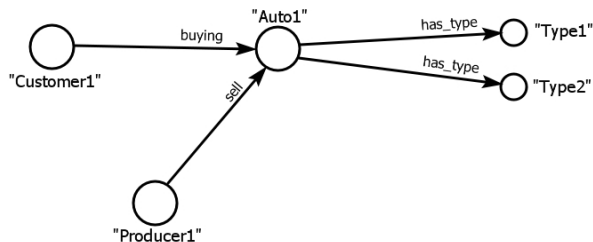


Рис. 2 Модель графовой базы данных

Необходимые SQL-запросы:

```

SELECT id, name from customers where
Customer customer=new Customer(rs.getInt("id"),
rs.getString("name"));
customer.save(); //создаем или обновляем узел
«Customer»; при создании индексируем по id.
    
```

```

SELECT id, name from auto where
Auto auto=new
Auto(rs.getInt("id"),rs.getString("name"));
auto.setType(rs.getString("type"));
auto.save(); //создаем или обновляем узел «Auto»;
создаем узел «type» если он не существует;
создаем отношение между «type» и «Auto»; при
создании индексируем по id.
    
```

```

SELECT id, name, type from merchants where
Producer producer=new Producer(rs.getInt("id"),
rs.getName("name"));
producer.save(); // создаем или обновляем узел
«Producer»; при создании индексируем по id.
    
```

```

SELECT customer_id,auto_id,buying_date from
customerBuying
Customer
customer=repository.getById(rs.getInt("customer_id
"));
customer.buyingAuto(rs.getInt("product_id"),rs.getDat
e("buying_date")); //создаем отношение между
«Customer» и «Auto» и устанавливаем дату
покупки как свойство связи.
    
```

После того как данные из реляционной базы импортированы и обе системы (MySQL и Neo4j) работают, становится актуальным вопрос синхронизации данных. В зависимости от поставленных задач синхронизацию можно проводить после конкретной операции, либо запускать по расписанию. Либо предусмотреть более сложную процедуру, которая будет отвечать необходимым требованиям и выполняться после очередного события.

Сравнение производительности Neo4j и MySQL

Поскольку структуры реляционной и графовой баз данных различны следует рассмотреть время поиска одной и той же информации в этих базах.

На рисунке 3 сравнение времени работы Neo4j и MySQL для поиска кратчайшего пути между двумя клиентами, при 5 различных вариантах (количество связей в базе). Ось Y представляет количество времени, которое занял каждый поиск кратчайшего пути, измеренный в миллисекундах (мс). Ось X представляет число связей, которые существовали в обеих база данных как в графовой так и в реляционной базе данных. В качестве испытываемого расстояние было выбрано расстояние между двумя узлами, которое наиболее часто встречалось в каждой из баз.

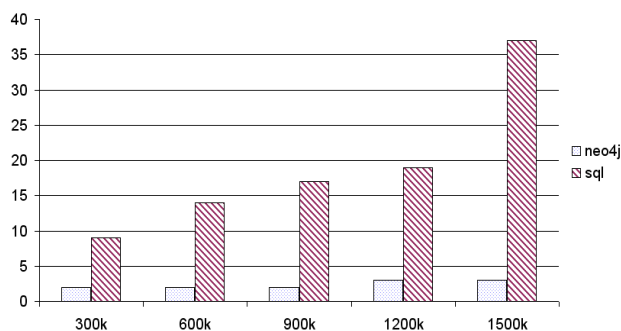


Рис. 3 Сравнение времени поиска кратчайшего расстояния между двумя узлами

Как видно по графику, система Neo4j значительно выигрывает по времени поиска кратчайшего пути для любого числа отношений, содержащихся в базе данных. Различия во времени обхода для MySQL непрерывно растут по сравнению с ростом числа отношений, которые существуют. Однако, можно видеть, что время поиска для Neo4j является почти постоянным. Вышеупомянутое наблюдение демонстрирует возможность Neo4j, и по-видимому других графовых баз данных, хорошо масштабироваться относительно размера базы данных.

На рисунке 4 отображена подробная производительность Neo4j во время экспериментальной оценки. Графически продемонстрировано среднее время в миллисекундах (мс) которое было затрачено для поиска кратчайший путь между любыми данными двух узлов во входных данных. Ось Y показывает потраченное время в миллисекундах. Ось X иллюстрирует расстояние между этими

двумя узлами, которое является числом транзитных участков, или количество узлов которые их соединяют. На графике представлены 5 видов кривых, которые соответствуют числу отношений, существующих в базе данных, в пределах от 300 000 - 1 500 000.

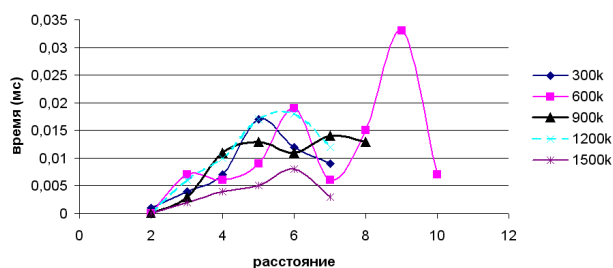


Рис. 4 Производительность Neo4j.

Рисунок 5 показывает подробную производительность для MySQL с теми же самыми входными данными как ранее описано для Neo4j. Данные снова показывают среднее время, в миллисекундах, которое необходимо для вычисления кратчайшего пути между любым данным двумя узлами во входных данных.

При помощи графика несложно убедиться в том, что производительность MySQL при нахождении кратчайшего пути ухудшается при увеличении числа отношений в базе данных. Это означает, что размер базы данных значительно влияет на производительность обходов пути, в отличие от того как это работает в Neo4j. Также видно, что с увеличением объема данных время обхода значительно увеличивается, в отличие от данных которые представлены в виде графовой базы Neo4j.

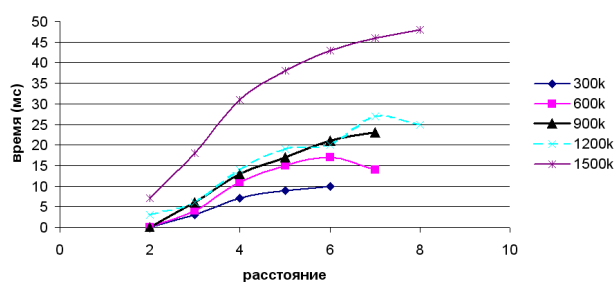


Рис. 5 Производительность MySQL.

Выводы

Таким образом, рассмотрен метод, который позволяет ускорить обработку информации в реляционной базе данных. При этом проблема использования данных различных типов из различных источников сводится к обычному парсингу. Хотя первичный импорт данных и может занять определенное время, также следует

учитывать и то, что это не частый процесс и данное время будет компенсировано за счет ускорения обработки информации в графовой структуре.

Используя данный подход получаем приложение которое отвечает потребностям современных приложений:

- улучшенная производительность разработки приложений за счет более гибкой модели данных;
- повышенная способность динамически масштабироваться для поддержки большего числа пользователей и данных;
- улучшенная производительность, которая позволит удовлетворить ожидания пользователей, желающих высокое быстродействие приложений и тем самым позволит выполнить более сложную обработку данных.

Список использованной литературы

1. *Jonas Partner, Aleksa Vukotic, Nicki Watt. Neo4j in Action / Jonas Partner, Aleksa Vukotic, Nicki Watt – Manning Publications, 2012. – 350p.*
2. *Maxim Kuznecov. MySQL 5 / M. Kuznecov M, Simdajnov I. – Spb.: BHV-Peterburg, 2006 – 1024p (in Russian).*
3. *Ian Robinson. Graph Databases / Ian Robinson, Jim Webber, Emil Eifrem – O'Reilly Media, 2013.– 178p.*
4. *Shashank Tiwari. Professional NoSQL / Shashank Tiwari –Wrox, 2011. – 408p.*
5. *Eric Redmond. Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement / Eric Redmond, D. R. Wilson [Jaclin Carter redaction / translate from English A.Slinkin] – M.: DMK Premm, 2013 – 384p. (in Russian).*
6. *Roy Goldman and Jennifer Widom. Dataguides: Enabling query formulation and optimization in semistructured databases. In Proceedings of the 23rd International Conference on Very Large Data Bases, VLDB '97, pages 436–445, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.*
7. *Pramod J. Sadalage; Martin Fowler. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence / Pramod J. Sadalage; Martin Fowler – Addison-Wesley Professional, 2012 – 192 p.*
8. *S. Melnik, H. Garcia-Molina, and E. Rahm, “Similarity flooding: a versatile graph matching algorithm and its application to schema matching, Proc. Intl. Conf. on Data Engineering, 2002, pp. 117–128.*

Надійшла до редколегії 20.06.13

УДК 004.655

Сенченко О.С., к.ф.-м.н., доц.

Закони де Моргана у табличних алгебрах

В роботі знайдено необхідні та достатні умови, за яких у табличних алгебрах виконуються закони де Моргана.

Ключові слова: бази даних, таблична алгебра, активне доповнення.

E-mail: senchenko@pisem.net

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

Вступ. В наш час системи керування базами даних широко використовуються в багатьох сферах діяльності людини. Найбільш розповсюдженою є реляційна модель даних, яка вперше була запропонована Е. Коддом у 1970 році [1]. З точки зору математики реляційна база даних є скінченною множиною скінченних відношень різної розмірності між заздалегідь визначеними множинами елементарних даних.

Табличні алгебри, що були введені В.Н. Редьком і Д.Б. Буєм, побудовані на основі реляційних алгебр Е. Кодда та суттєво їх уточнюють. Вони складають теоретичний фундамент мов запитів сучасних табличних баз даних. Елементи носія табличної алгебри уточнюють реляційні структури даних, а сигнатурні операції побудовані на базі основних табличних маніпуляцій в реляційних алгебрах та SQL-подібних мовах.

В монографії [2] знайдено велику кількість різних властивостей операцій табличних алгебр. У даній роботі знайдено та доведено необхідні та достатні умови, за яких деякі вclusions, які доведено в [2], перетворюються в рівності.

Основні визначення

Зафіксуємо деяку непорожню множину $A = \{A_1, \dots, A_n\}$, елементи якої називаються атрибутами. Довільну скінченну підмножину $R = \{A'_1, \dots, A'_k\} \subseteq A$ назовемо схемою, причому схема може бути порожньою множиною. Рядком s схеми R називається множина пар $s = \{(A'_1, d_1), \dots, (A'_k, d_k)\}$, проєкція якої по першій компоненті дорівнює R . Таблицею схеми R називається скінченна множина рядків схеми R . Далі в роботі розглядаємо таблиці схеми R з кількістю атрибутів k . На множині всіх таких таблиць введено операції об'єднання, перетину та різниці як аналогічні операції теорії множин;

A.S. Senchenko, PhD

De Morgan's laws in table algebra

In this paper is found the necessary and sufficient conditions due to which in table algebra De Morgan's laws are fulfilled.

Key Words: database, table algebra, active complement.

операції проєкції, селекції, з'єднання (в деяких джерелах, наприклад в [3], ця операція називається еквів'єднанням), ділення таблиць та операція перейменування атрибутів; ці операції не будуть використані в роботі, тому їх означення не наводимо. Також на множині таблиць введена операція активного доповнення, для визначення якої необхідно дати означення допоміжного поняття. Активним доменом атрибуту A відносно таблиці T називається множина $D_{A,T} = \{d \mid \exists s \in T \wedge (A, d) \in s\}$, яка, якщо казати змістовно, складається з усіх можливих значень атрибуту A в таблиці T .

Насиченням $C(T)$ називається таблиця $\prod_{A \in R} D_{A,T}$, де R - схема таблиці T , $\prod$ - оператор декартового добутку, що відповідає індексуванню за всіма атрибутами A схеми T . Активним доповненням таблиці T є таблиця $\tilde{T} = C(T) - T$.

Табличною алгеброю називають часткову алгебру з носієм - множиною всіх таблиць довільної схеми та наведеними вище дев'ятьма операціями (насичення є допоміжною операцією). В табличній алгебрі виділяють дві особливі таблиці: таблицю $T_\varepsilon = \{\varepsilon\}$, де ε - порожній рядок, схема якої є порожньою множиною та таблицю $T_\emptyset$ - порожню множину рядків довільної схеми.

Основні результати

В монографії [2] у підрозділі про властивості активного доповнення сформульовано та доведено ряд властивостей, більшість з яких є вclusions. Автором були знайдені необхідні та достатні умови, за яких два вclusions, що є аналогічними до відомих законів де Моргана, перетворюються у рівності, для особливих

таблиць рівності виконуються, але критерії можуть не виконуватися.

Теорема 1. Рівність $\tilde{T}_1 \cap \tilde{T}_2 = \left(T_1 \tilde{\cup} T_2 \right)$ при $\left(T_1 \tilde{\cup} T_2 \right) \neq T_\emptyset$ виконується тоді та лише тоді, коли виконується одна з чотирьох умов:

- 1) $\forall i D_{A_i, T_1} = D_{A_i, T_2}$.
- 2) $\forall i D_{A_i, T_2} \subseteq D_{A_i, T_1}$, та для усіх індексів q , $x \in D_{A_q, T_1} - D_{A_q, T_2}$ та усіх $d_1, \dots, d_{q-1}, d_{q+1}, \dots, d_k$, що належать активним доменам $D_{A_1, T_1}, \dots, D_{A_{q-1}, T_1}, D_{A_{q+1}, T_1}, \dots, D_{A_k, T_1}$, рядки $\{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, x), (A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\} \in T_1$.
- 3) $\forall i D_{A_i, T_1} \subseteq D_{A_i, T_2}$, та для усіх індексів q , $x \in D_{A_q, T_2} - D_{A_q, T_1}$ та усіх $d_1, \dots, d_{q-1}, d_{q+1}, \dots, d_k$, що належать активним доменам $D_{A_1, T_2}, \dots, D_{A_{q-1}, T_2}, D_{A_{q+1}, T_2}, \dots, D_{A_k, T_2}$, рядки $\{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, x), (A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\} \in T_2$.
- 4) Існує такий індекс q , для якого існують $x \in D_{A_q, T_1} - D_{A_q, T_2}$, $y \in D_{A_q, T_2} - D_{A_q, T_1}$ та $D_{A_q, T_2} \cap D_{A_q, T_1} \neq \emptyset$; для усіх $i \neq q$ виконуються рівності $D_{A_i, T_1} = D_{A_i, T_2}$; для усіх $x \in D_{A_q, T_1} - D_{A_q, T_2}$, усіх $y \in D_{A_q, T_2} - D_{A_q, T_1}$ та усіх $d_1, \dots, d_{q-1}, d_{q+1}, \dots, d_k$, що належать активним доменам $D_{A_1, T_1}, \dots, D_{A_{q-1}, T_1}, D_{A_{q+1}, T_1}, \dots, D_{A_k, T_1}$, рядки $\{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, x), (A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\} \in T_1$ і $\{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, y), (A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\} \in T_2$.

Доведення.

Доведемо спочатку достатність умов (1) – (4). У монографії [2] доведено включення $\tilde{T}_1 \cap \tilde{T}_2 \subseteq \left(T_1 \tilde{\cup} T_2 \right)$, тому нам потрібно довести

включення $\left(T_1 \tilde{\cup} T_2 \right) \subseteq \tilde{T}_1 \cap \tilde{T}_2$ при виконанні

кожної з умов (1) – (4) окремо від інших.

Для доведення достатності умови (1) доведемо спочатку допоміжне твердження.

Лема 1. Нехай виконуються рівності $\forall i D_{A_i, T_1} = D_{A_i, T_2}$. Тоді виконуються рівності $\forall i D_{A_i, T_1 \cup T_2} = D_{A_i, T_1} = D_{A_i, T_2}$.

Доведення лема 1. Оскільки $T_1 \subseteq T_1 \cup T_2$ та $T_2 \subseteq T_1 \cup T_2$, то за лемою 2.2.1 з [2] виконуються включення $\forall i D_{A_i, T_1} \subseteq D_{A_i, T_1 \cup T_2}$ та $\forall i D_{A_i, T_2} \subseteq D_{A_i, T_1 \cup T_2}$. Покажемо, що виконуються також обернені включення. Нехай для деякого індексу q виконується приналежність $d_q \in D_{A_q, T_1 \cup T_2}$, тоді існує такий рядок $s_q \in T_1 \cup T_2$, що $(A_q, d_q) \in s_q$. У цьому випадку виконується хоча б одна з умов $s_q \in T_1$ або $s_q \in T_2$. В першому випадку за визначенням активного домену виконується $s_q \in D_{A_q, T_1}$, в другому випадку виконується $s_q \in D_{A_q, T_2}$, і оскільки за умовою (1) $D_{A_q, T_1} = D_{A_q, T_2}$, то приналежності $s_q \in D_{A_q, T_1}$ та $s_q \in D_{A_q, T_2}$ виконуються одночасно. Тому виконуються включення $D_{A_q, T_1 \cup T_2} \subseteq D_{A_q, T_1}$ та $D_{A_q, T_1 \cup T_2} \subseteq D_{A_q, T_2}$. Із довільності індексу q випливає виконувальність включень $\forall i D_{A_i, T_1 \cup T_2} \subseteq D_{A_i, T_1}$ та $\forall i D_{A_i, T_1 \cup T_2} \subseteq D_{A_i, T_2}$, що доводить рівності $\forall i D_{A_i, T_1 \cup T_2} = D_{A_i, T_1} = D_{A_i, T_2}$. Лему 1 доведено.

Перейдемо тепер до доведення достатності умови (1): нехай виконуються рівності $\forall i D_{A_i, T_1} = D_{A_i, T_2}$. Від супротивного, припустимо, що включення не виконується. Тоді існує такий рядок $s = \{(A_1, d_1), \dots, (A_k, d_k)\}$, що $s \in \left(T_1 \tilde{\cup} T_2 \right)$

та $s \notin \tilde{T}_1 \cap \tilde{T}_2$. За визначенням операції активного доповнення виконуються приналежності $\forall i d_i \in D_{A_i, T_1 \cup T_2}$. Тоді з лема 1 випливає, що $\forall i d_i \in D_{A_i, T_1}$ та $\forall i d_i \in D_{A_i, T_2}$. Оскільки $s \notin T_1 \cup T_2$, то $s \notin T_1$ та $s \notin T_2$. Тоді за визначенням операції активного доповнення $s \in \tilde{T}_1$ та $s \in \tilde{T}_2$, тому $s \in \tilde{T}_1 \cap \tilde{T}_2$, що є протиріччям припущенню та доводить достатність умови (1).

Для доведення достатності умови (2), доведемо спочатку допоміжні твердження.

Лема 2. З включень $\forall i D_{A_i, T_2} \subseteq D_{A_i, T_1}$ випливають рівності $\forall i D_{A_i, T_1 \cup T_2} = D_{A_i, T_1}$.

Доведення лема 2. Оскільки $T_1 \subseteq T_1 \cup T_2$, то за лемою 2.2.1 з [2] виконуються включення $\forall i D_{A_i, T_1} \subseteq D_{A_i, T_1 \cup T_2}$. Обернені включення також

виконуються: нехай для деякого індексу q виконується приналежність $w_q \in D_{A_i, T_1 \cup T_2}$; отже, в таблиці $T_1 \cup T_2$ існує такий рядок w , що $(A_q, w_q) \in w$. За визначенням об'єднання таблиць рядок w повинен належати хоча б одній з таблиць T_1 або T_2 . Тоді в першому випадку за означенням активного домену $w_q \in D_{A_i, T_1}$, а в другому $w_q \in D_{A_i, T_2} \subseteq D_{A_i, T_1}$, тобто $w_q \in D_{A_i, T_1}$. Лему 2 доведено

Лема 3. За виконання умови (2) для будь-якого рядка $w = \{(A_1, w_1), \dots, (A_k, w_k)\} \in (T_1 \tilde{\cup} T_2)$ виконуються приналежності $w_1 \in D_{A_1, T_2}, \dots, w_k \in D_{A_k, T_2}$.

Доведення лема 3. Від супротивного припустимо, що $w_q \notin D_{A_q, T_2}$ для деякого індексу q . За визначенням активного домену виконується приналежність $w_q \in D_{A_q, T_1 \cup T_2}$, з лема 2 випливає, що $w_q \in D_{A_q, T_1}$, а з припущення $w_q \notin D_{A_q, T_2}$ випливає $w_q \in D_{A_q, T_1} - D_{A_q, T_2}$, тобто $s \in C(T_1)$ та $s \in C(T_2)$. Тоді з $w \in (T_1 \tilde{\cup} T_2)$ за визначенням операції активного доповнення випливає, що $w \notin T_1 \cup T_2$, тому $w \notin T_1$. Одержали, що існують $w_q \in D_{A_q, T_1} - D_{A_q, T_2}$ та $w_1, \dots, w_{q-1}, w_{q+1}, \dots, w_k$, які належать відповідно $D_{A_1, T_1}, \dots, D_{A_{q-1}, T_1}, D_{A_{q+1}, T_1}, \dots, D_{A_k, T_1}$, що $w = \{(A_1, w_1), \dots, (A_{q-1}, w_{q-1}), (A_q, w_q), (A_{q+1}, w_{q+1}), \dots, (A_k, w_k)\} \notin T_1$; це суперечить умові (2). Лему 3 доведено.

Перейдемо тепер до доведення достатності умови (2): нехай ця умова виконується. Від супротивного, припустимо, що включення не виконується. Тоді існує такий рядок

$s = \{(A_1, d_1), \dots, (A_k, d_k)\}$, що $s \in (T_1 \tilde{\cup} T_2)$ та

$s \notin \tilde{T}_1 \cap \tilde{T}_2$. З $s \in (T_1 \tilde{\cup} T_2)$ за визначенням

активного доповнення випливає $s \notin T_1 \cup T_2$, тобто $s \notin T_1$ та $s \notin T_2$. За лемою 3 виконуються приналежності $d_1 \in D_{A_1, T_2}, \dots, d_k \in D_{A_k, T_2}$, тому з умов $\forall i D_{A_i, T_2} \subseteq D_{A_i, T_1}$ виконуються також

приналежності $d_1 \in D_{A_1, T_1}, \dots, d_k \in D_{A_k, T_1}$. Тоді за визначенням операції активного доповнення $s \in \tilde{T}_1$ і $s \in \tilde{T}_2$, тому $s \in \tilde{T}_1 \cap \tilde{T}_2$, що є протиріччям припущенню та доводить достатність умови (2).

Достатність умови (3) доводиться аналогічно шляхом заміни індексів таблиць.

Для доведення достатності умови (4), доведемо спочатку допоміжне твердження.

Лема 4. За виконання умови (4) для будь-якого рядка $w = \{(A_1, w_1), \dots, (A_{q-1}, w_{q-1}), (A_q, z),$

$(A_{q+1}, w_{q+1}), \dots, (A_k, w_k)\} \in (T_1 \tilde{\cup} T_2)$ виконується приналежність $z \in D_{A_q, T_1} \cap D_{A_q, T_2}$.

Доведення лема 4. За визначенням операції активного доповнення $z \in D_{A_q, T_1 \cup T_2}$, що є можливим в одному з трьох випадків: а) $z \in D_{A_q, T_1}$ та $z \notin D_{A_q, T_2}$; б) $z \in D_{A_q, T_2}$ та $z \notin D_{A_q, T_1}$; в) $z \in D_{A_q, T_1}$ та $z \in D_{A_q, T_2}$. Покажемо, що випадки (а) та (б) не виконуються. Від супротивного, нехай виконується випадок (а), тобто $z \in D_{A_q, T_1} - D_{A_q, T_2}$. Тоді за умовою (4) рядок w повинен належати таблиці T_1 , і, отже, таблиці $T_1 \cup T_2$. Одержали, що $w \in (T_1 \tilde{\cup} T_2)$ та $w \in T_1 \cup T_2$, що є неможливим за визначенням операції активного доповнення. Аналогічно шляхом заміни індексів таблиць доводиться неможливість випадку (б), тобто виконується випадок (в), при якому $z \in D_{A_q, T_1} \cap D_{A_q, T_2}$. Лему 4 доведено.

Перейдемо тепер до доведення достатності умови (4): нехай ця умова виконується. Від супротивного, припустимо, що включення не виконується. Тоді існує такий рядок $s = \{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, z), (A_{q+1}, d_{q+1}), \dots,$

$(A_k, d_k)\}$, що $s \in (T_1 \tilde{\cup} T_2)$ та $s \notin \tilde{T}_1 \cap \tilde{T}_2$.

З $s \in (T_1 \tilde{\cup} T_2)$ за визначенням операції

активного доповнення випливає, що $s \notin T_1 \cup T_2$ та $d_1 \in D_{A_1, T_1 \cup T_2}, \dots, d_{q-1} \in D_{A_{q-1}, T_1 \cup T_2}, d_{q+1} \in D_{A_{q+1}, T_1 \cup T_2}, \dots, d_k \in D_{A_k, T_1 \cup T_2}$, а за лемою 4 випливає приналежність $z \in D_{A_q, T_1} \cap D_{A_q, T_2}$. З рівностей

активних доменів $D_{A_i, T_1} = D_{A_i, T_2}$ для усіх $i \neq q$ впливають приналежності $d_1 \in D_{A_1, T_1}, d_1 \in D_{A_1, T_2}, \dots, d_{q-1} \in D_{A_{q-1}, T_1}, d_{q-1} \in D_{A_{q-1}, T_2}, d_{q+1} \in D_{A_{q+1}, T_1}, d_{q+1} \in D_{A_{q+1}, T_2}, \dots, d_k \in D_{A_k, T_1}, d_k \in D_{A_k, T_2}$.

виконання $z \in D_{A_q, T_1} \cap D_{A_q, T_2}$ впливає $z \in D_{A_q, T_1}$ та $z \in D_{A_q, T_2}$. З $s \notin T_1 \cup T_2$ випливає, що $s \notin T_1$ та $s \notin T_2$. Таким чином за визначенням операції активного доповнення отримуємо $s \in \tilde{T}_1$ та $s \in \tilde{T}_2$, тобто $s \in \tilde{T}_1 \cap \tilde{T}_2$, що є протиріччям припущенню та доводить достатність умови (4).

Перейдемо тепер до доведення необхідності умов (1) – (4). Нехай виконується рівність

$$\tilde{T}_1 \cap \tilde{T}_2 = \left(T_1 \tilde{\cup} T_2 \right) \text{ та не виконується умова (1)}$$

$\forall i D_{A_i, T_1} = D_{A_i, T_2}$. Покажемо, що в цьому випадку повинна виконуватися одна з умов (2) – (4).

Рівності $\forall i D_{A_i, T_1} = D_{A_i, T_2}$ не виконуються тоді і лише тоді, коли існує такий індекс q , що $D_{A_q, T_1} \neq D_{A_q, T_2}$; ця нерівність можлива лише за умови, коли існує хоча б один елемент, який належить одній множині та не належить іншій, тобто можливі такі випадки:

а) існує такий $x \in D_{A_q, T_1}$, що $x \notin D_{A_q, T_2}$;

б) існує такий $y \in D_{A_q, T_2}$, що $y \notin D_{A_q, T_1}$;

в) випадки (а) і (б) виконуються одночасно.

Покажемо, що випадок (а) при невиконанні (в) призведе до виконання умови (2), що випадок (б) при невиконанні (в) – до виконання умови (3), а виконання випадку (в) призведе до виконання умови (4).

Нехай виконується випадок (а), тобто $\exists x | x \in D_{A_q, T_1} \text{ та } x \notin D_{A_q, T_2}$. Спочатку покажемо, що для всіх $i \neq q$ виконуються включення $D_{A_i, T_2} \subseteq D_{A_i, T_1}$. Від супротивного, припустимо, що ці включення не виконуються, тобто $D_{A_z, T_2} \not\subseteq D_{A_z, T_1}$ для деякого індексу z . Тоді існує такий елемент w , що $w \in D_{A_z, T_2}$ та $w \notin D_{A_z, T_1}$.

Оскільки таблиця $\left(T_1 \tilde{\cup} T_2 \right)$ за умовою теореми не дорівнює $T_\emptyset$, існують такі елементи $d_1, \dots, d_k$, що $s = \{(A_1, d_1), \dots, (A_k, d_k)\} \in \left(T_1 \tilde{\cup} T_2 \right)$. Тоді рядок

$s' = \{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, x), (A_{q+1}, d_{q+1}), \dots, (A_{z-1}, d_{z-1}), (A_z, w), (A_{z+1}, d_{z+1}), \dots, (A_k, d_k)\}$ належить

таблиці $\left(T_1 \tilde{\cup} T_2 \right)$ та не належить таблиці

$\tilde{T}_1 \cap \tilde{T}_2$. Дійсно, оскільки $x \in D_{A_q, T_1}$, то

$x \in D_{A_q, T_1 \cup T_2}$, оскільки $w \in D_{A_z, T_2}$, то $w \in D_{A_z, T_1 \cup T_2}$,

елементи $d_1, \dots, d_{q-1}, d_{q+1}, \dots, d_{z-1}, d_{z+1}, \dots, d_k$

належать активним доменам відповідних індексам атрибутів відносно таблиці $T_1 \cup T_2$, що впливає з належності рядку s таблиці

$\left(T_1 \tilde{\cup} T_2 \right)$, тобто $s' \in C(T_1 \cup T_2)$. При цьому,

оскільки $s' \notin T_1$ (зважаючи на $w \notin D_{A_z, T_1}$), та

$s' \notin T_2$ (зважаючи на $x \notin D_{A_q, T_2}$), то $s' \notin T_1 \cup T_2$,

тобто за визначенням активного доповнення

$s' \in \left(T_1 \tilde{\cup} T_2 \right)$. З другого боку, оскільки

$w \notin D_{A_z, T_1}$, то $s' \notin C(T_1)$, отже, $s' \notin \tilde{T}_1$, а оскільки

$x \notin D_{A_q, T_2}$, то $s' \notin C(T_2)$, тому, $s' \notin \tilde{T}_2$. Таким

чином $s' \notin \tilde{T}_1 \cap \tilde{T}_2$, тобто рівність

$\tilde{T}_1 \cap \tilde{T}_2 = \left(T_1 \tilde{\cup} T_2 \right)$ не виконується, що

суперечить умові. Тому припущення невірне, тобто для всіх $i \neq q$ виконуються $D_{A_i, T_2} \subseteq D_{A_i, T_1}$.

Покажемо тепер, що для всіх $x \in D_{A_q, T_1} - D_{A_q, T_2}$ та всіх $d_1, \dots, d_{q-1}, d_{q+1}, \dots, d_k$,

що належать відповідно $D_{A_1, T_1}, \dots, D_{A_{q-1}, T_1},$

$D_{A_{q+1}, T_1}, \dots, D_{A_k, T_1}$, повинна виконуватися

приналежність $\{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, x),$

$(A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\} \in T_1$. Від супротивного,

припустимо, що для деяких $d'_1, \dots, d'_{q-1}, d'_{q+1}, \dots, d'_k$

ця приналежність не виконується. За

визначенням насичення рядок $s' = \{(A_1, d'_1), \dots,$

$(A_{q-1}, d'_{q-1}), (A_q, x), (A_{q+1}, d'_{q+1}), \dots, (A_k, d'_k)\} \in C(T_1 \cup T_2)$

За припущенням $s' \notin T_1$, а з умови $x \notin D_{A_q, T_2}$

впливає $s' \notin T_2$, тобто $s' \notin T_1 \cup T_2$. За

визначенням активного доповнення з

$s' \in C(T_1 \cup T_2)$ та $s' \notin T_1 \cup T_2$ випливає

приналежність $s' \in \left(T_1 \tilde{\cup} T_2\right)$. З іншого боку, з $x \notin D_{A_q, T_2}$ випливає $s' \notin C(T_2)$, тому $s' \notin \tilde{T}_2$, тобто рівність $\tilde{T}_1 \cap \tilde{T}_2 = \left(T_1 \tilde{\cup} T_2\right)$ не виконується, що суперечить умові. Таким чином, припущення невірне, тобто для всіх $x \in D_{A_q, T_1} - D_{A_q, T_2}$ та всіх $d_1, \dots, d_{q-1}, d_{q+1}, \dots, d_k$, що належать відповідно $D_{A_1, T_1}, \dots, D_{A_{q-1}, T_1}, D_{A_{q+1}, T_1}, \dots, D_{A_k, T_1}$, виконується приналежність $\{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, x), (A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\} \in T_1$.

Таким чином одержали, що виконання випадку $x \in D_{A_q, T_1} - D_{A_q, T_2}$ приводить до виконанню умов $\forall i \neq q \ D_{A_i, T_2} \subseteq D_{A_i, T_1}$ та $\{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, x), (A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\} \in T_1$ для всіх $d_1, \dots, d_{q-1}, d_{q+1}, \dots, d_k$, що належать відповідно $D_{A_1, T_1}, \dots, D_{A_{q-1}, T_1}, D_{A_{q+1}, T_1}, \dots, D_{A_k, T_1}$. При цьому можливий один з двох взаємовиключних випадків:

a1) $D_{A_q, T_2} \subseteq D_{A_q, T_1}$;

a2) $D_{A_q, T_2} \not\subseteq D_{A_q, T_1}$.

Виконання випадку (a1) призведе до виконання умови (2) теореми. Виконання умови (a2) можливе тільки в тому випадку, коли існує такий y , що $y \in D_{A_q, T_2} - D_{A_q, T_1}$. Далі аналогічно шляхом заміни x на y та індексів таблиць можна довести, що $\forall i \neq q \ D_{A_i, T_1} \subseteq D_{A_i, T_2}$ та $\{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, y), (A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\} \in T_2$ для всіх $d_1, \dots, d_{q-1}, d_{q+1}, \dots, d_k$, що належать відповідно $D_{A_1, T_2}, \dots, D_{A_{q-1}, T_2}, D_{A_{q+1}, T_2}, \dots, D_{A_k, T_2}$, тобто $\forall i \neq q \ D_{A_i, T_1} = D_{A_i, T_2}$; в цьому випадку приходимо до виконання умови (4) теореми.

Аналогічно доводиться, що виконання випадку (б) приводить до виконання умови (3) або умови (4) теореми. Теорему доведено.

Теорема 2. Рівність $\left(T_1 \tilde{\cap} T_2\right) = \tilde{T}_1 \cup \tilde{T}_2$ при

$\left(T_1 \tilde{\cap} T_2\right) \neq T_\emptyset$ виконується тоді і лише тоді,

коли виконується одна з двох взаємовиключних умов:

1) для кожного індексу i виконуються рівності $D_{A_i, T_1} = D_{A_i, T_1 \cap T_2}$ та $D_{A_i, T_2} = D_{A_i, T_1 \cap T_2}$.

2) для кожного індексу q та кожного такого x , що $x \in D_{A_q, T_1} - D_{A_q, T_1 \cap T_2}$ та всіх $d_1, \dots, d_{q-1}, d_{q+1}, \dots, d_k$, що належать відповідно $D_{A_1, T_1}, \dots, D_{A_{q-1}, T_1}, D_{A_{q+1}, T_1}, \dots, D_{A_k, T_1}$ рядки $\{(A_1, d_1), \dots, (A_{q-1}, d_{q-1}), (A_q, x), (A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\} \in T_1$. Для кожного індексу w , кожного такого y , що $y \in D_{A_w, T_2} - D_{A_w, T_1 \cap T_2}$ та всіх $d_1, \dots, d_{w-1}, d_{w+1}, \dots, d_k$, що належать відповідно $D_{A_1, T_2}, \dots, D_{A_{w-1}, T_2}, D_{A_{w+1}, T_2}, \dots, D_{A_k, T_2}$ рядки $\{(A_1, d_1), \dots, (A_{w-1}, d_{w-1}), (A_w, y), (A_{w+1}, d_{w+1}), \dots, (A_k, d_k)\} \in T_2$.

Доведення.

Покажемо спочатку достатність умов (1) та (2). В монографії [2] доведено включення $\left(T_1 \tilde{\cap} T_2\right) \subseteq \tilde{T}_1 \cup \tilde{T}_2$, тому нам потрібно довести

включення $\tilde{T}_1 \cup \tilde{T}_2 \subseteq \left(T_1 \tilde{\cap} T_2\right)$ при окремому виконанні умов (1) та (2). Зауважимо, що з $\left(T_1 \tilde{\cap} T_2\right) \neq T_\emptyset$ випливає $\tilde{T}_1 \cup \tilde{T}_2 \neq T_\emptyset$.

Доведемо достатність умови (1). Нехай виконуються рівності $\forall i \ D_{A_i, T_1} = D_{A_i, T_1 \cap T_2}$ та $\forall i \ D_{A_i, T_2} = D_{A_i, T_1 \cap T_2}$. Від супротивного, припустимо, що включення не виконується, тобто існує такий рядок, $s = \{(A_1, d_1), \dots, (A_k, d_k)\} \in \tilde{T}_1 \cup \tilde{T}_2$ що $s \notin \left(T_1 \tilde{\cap} T_2\right)$.

З приналежності $s \in \tilde{T}_1 \cup \tilde{T}_2$ випливає виконання хоча б однієї з приналежностей: $s \in \tilde{T}_1$ або $s \in \tilde{T}_2$. Нехай $s \in \tilde{T}_1$. За визначеннями насичення та активного доповнення $d_1 \in D_{A_1, T_1}, \dots, d_k \in D_{A_k, T_1}$ та $s \notin T_1$. З умови (1) випливають приналежності $d_1 \in D_{A_1, T_1 \cap T_2}, \dots, d_k \in D_{A_k, T_1 \cap T_2}$, тобто $s \in C(T_1 \cap T_2)$. За визначенням операції перетину з $s \notin T_1$ випливає $s \notin T_1 \cap T_2$. Тоді за визначенням активного доповнення $s \in \left(T_1 \tilde{\cap} T_2\right)$, що суперечить припущенню, тому включення

$\tilde{T}_1 \cup \tilde{T}_2 \subseteq \left(T_1 \tilde{\cap} T_2 \right)$ виконується. Аналогічно

доводиться виконуватись включення при $s \in \tilde{T}_2$.

Для доведення достатності умови (2), доведемо спочатку допоміжне твердження.

Лема 5. При виконанні умови (2) для будь-якого рядка $s = \{(A_1, d_1), \dots, (A_k, d_k)\} \in \tilde{T}_1 \cup \tilde{T}_2$ виконуються приналежності $d_1 \in D_{A_1, T_1 \cap T_2}, \dots, d_k \in D_{A_k, T_1 \cap T_2}$.

Доведення лема 5. Від супротивного, припустимо, що твердження лема не виконується, тобто $d_q \notin D_{A_q, T_1 \cap T_2}$ для деякого індексу q . З

приналежності $s \in \tilde{T}_1 \cup \tilde{T}_2$ випливає виконання хоча б однієї з приналежностей: $s \in \tilde{T}_1$ або $s \in \tilde{T}_2$.

Нехай виконується $s \in \tilde{T}_1$. Тоді за визначенням активного доповнення $s \notin T_1$ та виконуються приналежності $d_1 \in D_{A_1, T_1}, \dots, d_k \in D_{A_k, T_1}$, тобто $d_q \in D_{A_q, T_1} - D_{A_q, T_1 \cap T_2}$. За умовою (2) в цьому випадку повинна виконуватись приналежність $s \in T_1$; протиріччя доводить неможливість випадку $s \in \tilde{T}_1$. Нemoжливість $s \in \tilde{T}_2$ доводиться аналогічно.

Перейдемо тепер до доведення достатності умови (2). Від супротивного, припустимо, що включення не виконується, тобто існує такий рядок $s = \{(A_1, d_1), \dots, (A_k, d_k)\} \in \tilde{T}_1 \cup \tilde{T}_2$, що

$s \notin \left(T_1 \tilde{\cap} T_2 \right)$. За лемою 5 виконуються

приналежності $d_1 \in D_{A_1, T_1 \cap T_2}, \dots, d_k \in D_{A_k, T_1 \cap T_2}$,

тому за означенням насичення $s \in C(T_1 \cap T_2)$. З

приналежності $s \in \tilde{T}_1 \cup \tilde{T}_2$ випливає $s \in \tilde{T}_1$ або

$s \in \tilde{T}_2$, тому $s \notin T_1$ або $s \notin T_2$; у будь-якому

випадку $s \notin T_1 \cap T_2$. За визначенням активного

доповнення $s \in \left(T_1 \tilde{\cap} T_2 \right)$, що суперечить

припущенню та доводить достатність умови (2).

Перейдемо тепер до доведення необхідності умов (1) і (2). Нехай виконується рівність з формулювання теореми та не виконується умова (1). Покажемо, що у такому випадку повинна виконуватись умова (2).

Рівності $\forall i D_{A_i, T_1} = D_{A_i, T_1 \cap T_2}$ та

$\forall i D_{A_i, T_2} = D_{A_i, T_1 \cap T_2}$ не виконуються тоді і лише

тоді, коли існує такий індекс q , що

$D_{A_q, T_1} \neq D_{A_q, T_1 \cap T_2} \vee D_{A_q, T_2} \neq D_{A_q, T_1 \cap T_2}$. Припустимо,

що $D_{A_q, T_1} \neq D_{A_q, T_1 \cap T_2}$; ця нерівність зважаючи на

включення $T_1 \cap T_2 \subseteq T_1$ може виконуватись лише

в тому випадку, коли $\exists x \in D_{A_q, T_1} - D_{A_q, T_1 \cap T_2}$. Від

супротивного, припустимо, що умова (2) не

виконується, тобто існують такі $d_1, \dots, d_{q-1}$,

$d_{q+1}, \dots, d_k$, які належать відповідно $D_{A_1, T_1}, \dots,$

$D_{A_{q-1}, T_1}, D_{A_{q+1}, T_1}, \dots, D_{A_k, T_1}$, що рядок $s = \{(A_1, d_1), \dots,$

$(A_{q-1}, d_{q-1}), (A_q, x), (A_{q+1}, d_{q+1}), \dots, (A_k, d_k)\}$ не

належить таблиці T_1 . Тоді оскільки $x \notin D_{A_q, T_1 \cap T_2}$,

то $s \notin C(T_1 \cap T_2)$, і тому, $s \notin \left(T_1 \tilde{\cap} T_2 \right)$. З іншого

боку, $s \in C(T_1)$ і $s \notin T_1$, отже, $s \in \tilde{T}_1$, тому

$s \in \tilde{T}_1 \cup \tilde{T}_2$, це суперечить рівності

$\left(T_1 \tilde{\cap} T_2 \right) = \tilde{T}_1 \cup \tilde{T}_2$. Це протиріччя показує, що при

$D_{A_q, T_1} \neq D_{A_q, T_1 \cap T_2}$ умова (2) повинна

виконуватись. Той факт, що з нерівності

$D_{A_q, T_2} \neq D_{A_q, T_1 \cap T_2}$ повинна виконуватись умова (2)

доводиться аналогічно. Теорему доведено.

Висновки

В роботі знайдено критерії, за яких у

табличних алгебрах виконуються закони де

Моргана. Результати роботи можуть бути

використаними в теорії узагальнених табличних

алгебр та, на наш погляд, для оптимізації запитів

в реляційних базах даних.

Список використаних джерел

1. Codd E.F. A Relational Model of Data for Large Shared Data Banks / E. F. Codd // Communications of the ACM. – 1970. – V. 13, N. 6. – P. 377–387.
2. Relational data base: table algebras and family of the SQL languages / [V.N.Redko, U.I.Brona, D.B.Buy, S.A.Polyakov]. – Kiev: Publisher «Academperiodica», 2001. – 198 p. (in Ukrainian).
3. D. Maier. The theory of Relational Databases: [translation from English] / D. Maier. – Moscow: «Mir», 1987. – 608 p. (in Russian).

Надійшла до редколегії 27.06.13

УДК 004.4

Слабоспицька О.О.¹, к.ф.-м.н., с.н.с.

Розвиток моделі властивостей сімейства програмних продуктів для забезпечення їх адаптовності

Модель властивостей Сімейства програмних продуктів розвинуто об'єктно-компонентним методом К.М.Лавріщевої-В.М.Грищенка для забезпечення адаптовності продуктів до змін їх функцій і даних, передбачених у моделі й довільних. Нова модель – пара ізоморфних алгебраїчних моделей. Їх носії – множини об'єктів, інтерфейси яких підтримують функції з даними, і компонентів повторного використання для реалізації інтерфейсів. Відношення подають передбачені зміни функцій і даних. Надано операції добору підмножини об'єктів і компонентної реалізації відповідних їм функцій у продуктах. За цією моделлю запропоновано об'єктно-компонентну модель продукту та алгебри його структурного рефакторингу й реінжинірингу з операціями передбаченої й довільної зміни. Окреслено формальні засоби використання моделей Сімейства/продукту і алгебр.
Ключові слова: сімейство програмних продуктів, модель властивостей, адаптовність, об'єктно-компонентний метод, ізоморфізм.

¹. Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т Глушкова, 4д, e-mail: ols.07@mail.ru

O.A. Slabospickaya¹, Cand. Sci. (Phys.–Math.)

Feature Model of Software Product Line Enhancing to Enable Product Adaptability

Feature Model of Software Product Line is enhanced with E.Lavrysheva-V.Grischenko's Object-Component Method to enable products adaptability for their functions and data changes either predicted in the model or not. New model is a pair of isomorphic algebraic models. Their bases are the sets of objects supporting the functions with data through interfaces and respectively of reusable components implementing interfaces. Relations represent functions and data predicted changes. Operations are contributed for objects subsets choice and corresponding functions component implementation in the products. Based on this model, object-component model for sample product and algebras for its structured refactoring and reengineering with the operations of predicted and anywise changes are proposed. Formal means are drafted for Product Line/product models as well as new algebras to use.

Key Words: software product line, feature model, adaptability, object-component method, isomorphism.

¹. Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova st., 4d, e-mail: ols.07@mail.ru.

Статтю представив д.ф.-м.н., проф. Буй Д.Б.

Вступ

Перспективною формою індустріального виробництва програмних продуктів (ПП) масового використання з гарантованою якістю, низькими витратами й стислими термінами є наразі Сімейство ПП (СПП). Це набір ПП, що розробляються попередньо визначеним способом із спільної множини ресурсів повторного використання і мають керовану множину спільних властивостей (features), відповідних потребам споживачів у цільовій предметній області (ПрО) [1]. Окремий ПП – програмна система з сертифікатом

автентичності та експлуатаційною документацією. Властивість – вимога, функція чи нефункціональна характеристика, запитана споживачами.

Визначальними особливостями СПП є [1]:

- процеси конструювання домену й застосунків, які продукують ресурси та, відповідно, ПП;
- простори Проблеми й Рішень, утворені передбаченими властивостями ПП, та, відповідно, ресурсами і правилами їх композування у ПП.

Стандарт “де-факто” опису простору Проблеми – модель властивостей (МВ) [1,2]. Це ієрархія змінних та спільних властивостей ПП із спеціальними відношеннями [2,3] їх взаємозалежно-

сті. Вона формує передумови переваг СПП: гарантування якості, адаптованості до передбачених змін потреб споживачів і умов використання, скорочення витрат і термінів розробки ПП. Однак їх досягненню перешкоджає обмеження МВ – відсутність формалізмів зв'язування з властивостями і композування програмних ресурсів.

Традиційні засоби реалізації зв'язування потребують змістовного ручного коригування коду для побудови ПП з непередбаченими властивостями та еволюції СПП. Воно знижує якість ресурсів і ПП та збільшує час/вартість їх тестування, додатково обмежуючи переваги СПП. Тому обмеження МВ стають критичними у динамічних слабо формалізованих Про, що вимагають постійних змін ПП й еволюції СПП.

Аналіз підходів до підвищення ефективності повторного використання коду засвідчує перспективність опрацювання обмежень МВ розглядом ресурсу як компонента повторного використання (КПВ). Для його реалізації д. ф.-м. н., проф. К.М. Лавріщевою запропоновано застосування до СПП об'єктно-компонентного методу проектування програмних систем (ОКМ) створеного й апробованого [4-8] нею у співавторстві з В.М.Грищенком (2003-2012рр.).

Розвиток МВ на допомогу ОКМ і розробка підходу до використання отриманої об'єктно-компонентної моделі для побудови СПП є метою роботи. Виклад ґрунтується на результатах проекту ІПС НАНУ ДР 0107U002205 «Розробка теоретичного фундаменту генеруючого програмування та інструментальних засобів його підтримки» під керівництвом д-ра фіз.-мат. наук, проф. К.М. Лавріщевої [4, 5]. Надана модель конструктивно уточнює модель варіабельності в структурі СПП, розроблену за участі автора [9].

Автор висловлює щире подяку проф. К.М. Лавріщевій за рекомендацію ОКМ для розвитку МВ та плідні зауваження щодо змісту роботи.

Об'єктно-компонентна модель СПП

Для побудови об'єктно-компонентної моделі ОКМ застосовано до МВ з властивостями двох типів – передбаченими функціями ПП та їх даними. Зміст, етапи і результати застосування ОКМ показано на рис. 1а).

Визначальний елемент формованої моделі – об'єктна підмодель СПП (ОА). Для її побудови чотири рівні об'єктного аналізу ОКМ [4-6] адаптовано до МВ. Сутність адаптації подано на рис. 1б) і в табл. 1 (“прапорці” позначають поповнення ОКМ).

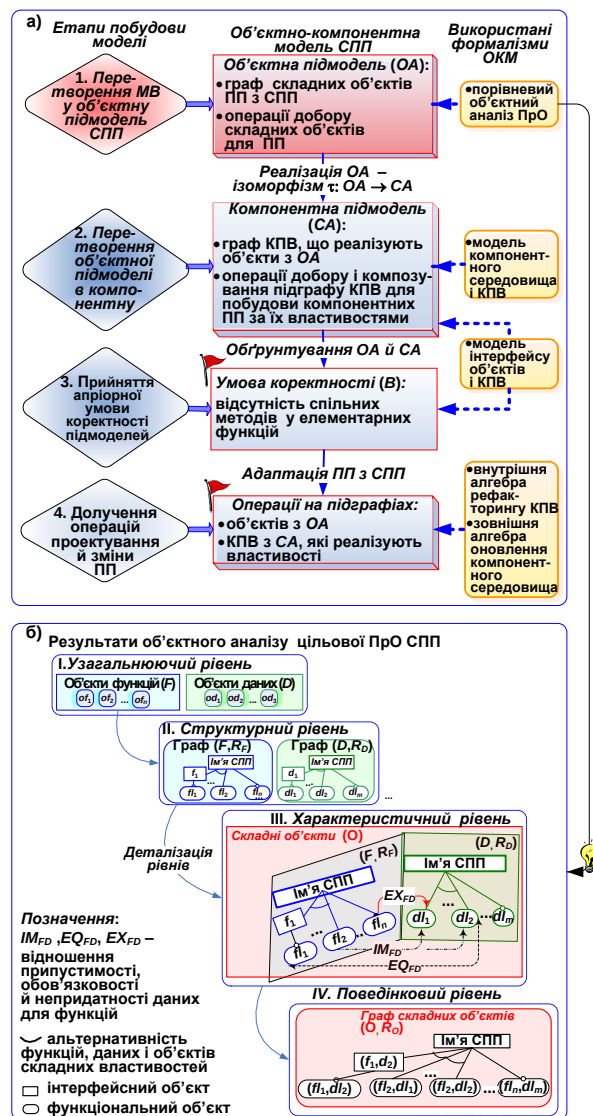


Рис.1. Сутність розвитку ОКМ для СПП

Згідно з рис. 1б), модель ОА – граф складних об'єктів $o \in O$, інтерфейси яких подають усі передбачені функції ПП і дані, що можуть або мають використовуватися ними. Об'єкти даних і функції пов'язані відношеннями IM_{FD} й EQ_{FD} (див. рис. 1б)), що формалізують ім-плікацію й еквівалентність між ними як властивостями в МВ. При цьому тільки термінальні об'єкти (позначені на рис. 1б) овалами) є функціональними [7, 8]: їх інтерфейси містять предметно-орієнтовані методи. Неперетинність методів таких об'єктів – апіорна умова завершення об'єктного аналізу. Вузли ж графа (прямокутники на рис. 1б)) є інтерфейсними об'єктами [7, 8]. Вони містять методи виклику предметно-орієнтованих методів функціональних об'єктів, підтримуючи їх взаємодії, подані на рис.1б) дугами графа (R_O). Вигляд отриманої моделі фіксує

Таблиця 1

Розвиток об'єктного аналізу в ОКМ для СПП

Рівень	Канонічне подання	Подання для МВ СПП
Узагальнюючий	Множина об'єктів Про	Множини об'єктів функцій (F) і даних (D)
Структурно-впорядковуючий	Граф об'єктів, де дуги подають відношення "частка-ціле" між ними	Графи об'єктів функцій і даних, де дуги (R_F, R_D) подають спеціальні синтаксичні відношення їх взаємозалежності у МВ [2,3], а списки методів для елементарних функцій неперетинні
Характеристичний	Граф об'єктів, доповнених унарними предикатами їх характеристик	Множина складних об'єктів (O) – пар об'єктів функцій і даних, що можуть або мають використовуватися функціями за МВ, інтерфейси яких – кортежі сигнатур методів функцій і атрибутів даних
Поведінковий	Граф переходів між станами об'єктів	Граф складних об'єктів (O, R_O), де дуги подають відношення передавання даних між ними згідно з відношеннями взаємозалежності в ПП, індукованими R_F, R_D , а списки методів для листків неперетинні

Означення 1. Об'єктно-компонентна модель (ОКМ-модель) СПП – це п'ятірка

$$\begin{aligned}
 OCM &= \langle r; OA; CA; B; GP; CP \rangle; \\
 OA &= \langle O, R_O, OO \rangle; CA = \langle C, R_C, CO \rangle; \\
 R_O &= \{CH_O, VC_O, IM_O, EQ_O, EX_O\}; \\
 R_C &= \{CH_C, VC_C, IM_C, EQ_C, EX_C\}; \\
 \forall E \in R_O \ E &\subset O \times O; \forall E \in R_C \ E \subset C \times C,
 \end{aligned} \quad (1)$$

де r – ім'я СПП;

OA, CA – об'єктна й компонентна підмоделі;

B – апріорна умова коректності;

GP – операції первинного проектування ПП;

CP – операції зміни певного ПП з СПП.

У підмоделях OA й CA з (1) O і C – множини складних об'єктів, інтерфейси яких подають усі передбачені функції ПП з їх даними, та КПВ, що реалізують ці інтерфейси; R_O і R_C – конкретизації для O і C відношень взаємозалежності елементів МВ [2,3], які описують обмін даних між об'єктами та змінні взаємодії КПВ у ПП; OO і CO – операції добору й, відповідно, реалізації припустимих підмножин складних об'єктів для компонентних ПП з СПП. Ім'я r є спільним коренем графів (O, R_O) і (C, R_C).

Складний об'єкт з (1) має вигляд

$$o = (f, d(f)) \mid fIM_{FD}d(f) \mid fEQ_{FD}d(f) \subseteq F \times D, \quad (2)$$

та інтерфейс

$$in(o) = \langle id, M(f), D(d(f)) \rangle \in I, \quad (3)$$

де $f \in F$ і $d(f) \in D$ – об'єкти функції та, відповідно, даних, які f може чи має використовувати в ПП;

$IM_{FD}, EQ_{FD} \subseteq F \times D$ – відношення можливості й обов'язковості даних для функції;

id – унікальне ім'я in , $M(f)$ – сигнатура методів f ; $D(d(f))$ – список атрибутів $d(f)$, зіставлених змінним з сигнатур методів $M(f)$.

Умова коректності моделі OCM (B в (1)) – завершення для неї об'єктного аналізу, тобто відсутність спільних методів елементарних функцій.

Множини відношень R_O і R_C з (1) описує

Означення 2. Функції $f_i, f_j \in F$ пов'язані:

1) прямим (CH_F) або (m, n) -варіантним підпорядкуванням ($VC_F^{m,n}, 0 \leq m < n < |F|$), якщо реалізація в ПП підпорядковуючої функції f_i впливає з реалізації всіх чи, відповідно, від m до n з $VF \subset F$ підпорядкованих функцій f_j ;

2) імплікацією IM_F чи зіставленням EQ_F , якщо реалізація f_i потребує f_j або, відповідно, вони можуть реалізуватися тільки разом;

3) виключенням EX_F , якщо реалізація f_i внеможливіє реалізацію f_j , причому IM_F, EQ_F, EX_F неприпустимі для функцій, спільно підпорядкованих одній функції.

Відношення між об'єктами даних мають той же зміст, що й відношення 1)-3), але $VC_F^{m,n}$ відповідає $VC_D^{p,q}, 0 \leq p < q < |D|$.

Складні об'єкти $o_i = (f_i, d(f_i)), o_j = (f_j, d(f_j)) \in O$ (2) пов'язані прямим підпорядкуванням (CH_O), імплікацією (IM_O), зіставленням (EQ_O) і виключенням (EX_O), якщо однойменні відношення на F (CH_F, IM_F, EQ_F, EX_F) пов'язують функції $f_i, f_j \in F$ або ж f_i, f_j однакові/непов'язані, а однойменні відношення на D (CH_D, IM_D, EQ_D, EX_D) мають місце для $d(f_i), d(f_j)$. Відношення (m, n, p, q) -варіантного підпорядкування $VC_O^{m,n,p,q}$ має місце, якщо $VC_F^{m,n}$ пов'язує f_i, f_j ($p = q = 0$) або f_i, f_j однакові/непов'язані, і $VC_D^{p,q}$ пов'язує $d(f_i), d(f_j)$ ($m = n = 0$).

Відношення з множини R_C та однойменні відношення на множині інтерфейсів $I = \{in(o)\}$ (3)

$$R_I = \{CH_I, VC_I^{m,n,p,q}, IM_I, EQ_I, EX_I\} \quad (4)$$

мають місце для реалізуючих КПВ (відповідно, інтерфейсів), якщо їх однойменні відповідники з R_O пов'язують підтримувані об'єкти o_i, o_j .

Запроваджені відношення R_O, R_I, R_C дозволяють формалізувати для ОКМ-моделі поняття

точки варіабельності – “здатності ПП або артефакта до ефективного розвитку, зміни, налаштування чи конфігурування для використання в певному контексті” [1-3] та варіанта.

Означення 3. Точкою варіабельності складних об’єктів, інтерфейсів або КПВ зветься об’єкт (інтерфейс чи КПВ) з непорожньою множиною $VR(m, n, p, q)$ -варіантно підпорядкованих об’єктів (відповідно, інтерфейсів або КПВ). Підпорядковані об’єкти (інтерфейси, КПВ) зветься варіантами реалізації точки в ПП. Об’єкти, інтерфейси й КПВ, що не є варіантами, зветься обов’язковими.

Зауваження 1. Варіант з VR для точки варіабельності в O, I, C має один з базових типів t :

- опціональний ($t=1; m=0, n=1$), що може як реалізуватися в ПП, так і бути відсутнім;
- альтернативний ($t=2; m=n=1$), який має реалізуватися в ПП тільки один;
- варіантний ($t=3; m=1, n \geq 1$) – має реалізуватися хоча б один довільний елемент VR ;
- опціонально-варіантний ($t=4; m=0, n \geq 1$) – має реалізуватися довільна множина $\emptyset \subseteq VI \subseteq VR$.

Зауваження 2. З урахуванням означень 2, 3, інтерфейс об’єкта-вузла в графі (O, R_O) можна подати кортежем сигнатур методів виклику методів для об’єктів, підпорядкованих o ($MC(o)$), з їх даними ($DC(o)$) та сигнатури $mch(o)$ сервісного методу вибору варіантів o для реалізації:

$$\varepsilon(o) = in(o) = \langle id, MC(o), mch(o), DC(o) \rangle. \quad (5)$$

Точки варіабельності є “центрами” її локалізації в ПП й артефактах СПП та “мішенями” передбачених в СПП змін, межі яких фіксуються варіантами. Отже, запропонована ОКМ-модель СПП OSM (1) є фактично моделлю локалізації варіабельності в його структурі. Вона обумовлює вигляд похідних ОКМ-моделей: локалізації варіабельності в ПП з СПП й оцінювання її рівня, запропонованих далі для забезпечення адаптивності ПП й еволюції СПП у процесі його об’єктно-компонентного розроблення. Кожна з них конструктивно уточнює, в парадигмі збіркового програмування, однойменну високорівневу модель [9, 10], розроблену без врахування способу реалізації/застосування програмних ресурсів.

Вигляд КПВ $c \in C$ з CA в (1) на підставі моделі КПВ в ОКМ [4-6] і R_O, R_I (4), R_C встановлює

Означення 4. Об’єкт $o = (f, d(f)) \in O$ (2) реалізується відображенням

$$\tau: O \rightarrow C = \{c = \tau(o)\}, \quad (6)$$

що зіставляє об’єкту o КПВ

$$c = \tau(o) = \langle id, II.o, S.o \cup ch.o, IO.o \rangle, S.o \cup ch.o \neq \emptyset, \quad (7)$$

який має:

- 1) унікальне ім’я id , успадковане від інтерфейсу $in(o)$ об’єкта o , що реалізується;
- 2) єдиний вхідний інтерфейс $II.o$ – інтерфейс підтримуваного об’єкта $in(o)$ (3), (5);
- 3) множини $S.o$ програмних реалізацій $M(f)$ для атрибутів $D(d(f))$ і $ch.o$ – методу $mch(o)$ у (5);
- 4) множину вихідних інтерфейсів $IO.o$, яка утворена інтерфейсами $in(o^*)$ (3) об’єктів $o^* \in O$, підпорядкованих o в (O, R_O) :

$$IO.o = \bigcup_{\substack{0 \leq m < n < |F|, \\ 0 \leq p < q < |D|}} \{in(o^*) \in O \mid oVC_o^{m,n,p,q} o^*, oCH_o o^*\}. \quad (8)$$

У виразі (7) системні сервіси й сервіси управління екземплярами КПВ вилучено з його ОКМ-моделі [4-6] для спрощення нотації.

3 виразів (5), (7), (8) та означення 2 впливає

Лема. Множина (8) вихідних інтерфейсів КПВ $\tau(o)$ (6), що реалізує об’єкт $o \in O$, утворена вхідними інтерфейсами КПВ c^* , підпорядкованих $\tau(o)$ в графі (C, R_C) , або ж інтерфейсами ir , підпорядкованими інтерфейсу o $in(o)$ в (I, R_I) з R_I (4):

$$\begin{aligned} IO.o &= \bigcup_{\substack{0 \leq m < n < |F|, \\ 0 \leq p < q < |D|}} \{II.c^* \in C \mid \tau(o)VC_c^{m,n,p,q} c^*, \tau(o)CH_c c^*\}; \quad (9) \\ &= \bigcup_{\substack{0 \leq m < n < |F|, \\ 0 \leq p < q < |D|}} \{ir \in I \mid in(o)VC_I^{m,n,p,q} ir, in(o)CH_I ir\}. \quad (10) \end{aligned}$$

Вхідний інтерфейс $II.o$ має три подання “покоординатним” (тобто для сигнатур методів виклику функцій і елементарних методів $M(f)$, сигнатур методів mch та атрибутів даних) об’єднанням: інтерфейсів $in(o^*)$ об’єктів, підпорядкованих o^* , з виразу (8); вхідних інтерфейсів $II.c^*$ КПВ, підпорядкованих $\tau(o)$, з (9); інтерфейсів ir з (10).

Наслідок. Для КПВ (7) виконуються надані ОКМ умови взаємодії [4-6] підпорядковуючого й підпорядкованого КПВ у графі (C, R_C) з (1):

$$\forall c, c^* \in C, 0 \leq m < n < |F|, 0 \leq p < q < |D| \quad (11) \\ (cCH_c c^* \vee cVC_c^{m,n,p,q} c^*) \Rightarrow II.o^* \in IO.o$$

та цілісності: єдиний вхідний інтерфейс $II.o$ (7) має хоча б одну реалізацію з $S.o \cup ch.o$.

Для визначення операцій в OSM (1) запровадимо поняття конфігурації елементів множин O і C за допомогою

Означення 5. В СПП з моделлю (1) конфігурація об’єктів – зв’язний підграф $g = (PO; R_O)$, $PO \subseteq O$ графу $(O; R_O)$ з коренем r , де вершини не пов’язані виключенням і зберігають свої відношення з $(O; R_O)$. Конфігурація КПВ – подібний до g підграф $q(C)$ графу (C, R_C) з (1).

Прийняте означення дозволяє формалізацію компонентного ПП з СПП для створення формального апарату підтримки зміни ПП.

Означення 6. ОКМ-модель ПП з унікальним іменем ip в СПП (1) – трійка

$$PM = \langle ip; g; [\tau(g)] \rangle, \tau(g) = \{c = \tau(o) \in C, o \in SO\}, \quad (12)$$

де g – конфігурація складних об'єктів для ПП.

Сам ПП – сукупність взаємодіючих КПВ, або ж їх збірка, з образу конфігурації об'єктів ПП.

У СПП (1) ПП з моделлю $nm = \langle ip; oc; cc \rangle$ вигляду (12), де необов'язково $cc = \tau(oc)$, є:

а) непередбаченим, якщо nm містить непередбачені об'єкти $NO \not\subset O$, не всі обов'язкові об'єкти або не відповідає R_O ;

б) непередбаченим за КПВ, якщо умови а) не виконуються, але nm невідповідна R_C і/або включає нові КПВ $NC \not\subset C$ чи не всі обов'язкові КПВ;

в) передбаченим за невиконання умов а), б).

Два ПП є близькими (відповідно, далекими) варіантами один одного, якщо в їх моделях обов'язкові об'єкти однакові (відповідно, перетинаються), але конфігурації об'єктів різні.

Модель (12) явно подає точки варіабельності та їх варіанти в окремому ПП, конструктивно деталізуючи для збіркового програмування високорівневу модель варіабельності в артефактах [9,10]. Завдяки цьому вона створює контекст проектування й адаптування ПП: визначає їх множину; виокремлює ПП з точками варіабельності, придатні до передбачених змін; зіставляє їм множини передбачених варіантів. Подання (12) дозволяє також запровадження метризованого відношення варіантності передбачених і непередбачених варіантів. Воно перспективне для створення методів автоматизованої перевірки приналежності ПП з моделлю (12) до СПП (1) та оптимізації конфігурації КПВ у ПП. Нарешті, припустимість у (12) точок варіантності уможлиблює не тільки зміну ПП часу розроблення, але й автоматизоване on-line-налаштування при запуску й навіть під час виконання.

Операції добору складних об'єктів (ОО) та, відповідно, їх реалізації в компонентних ПП з СПП (СО) є базовими в ОКМ-моделі ОСМ (1), являючи собою примітиви для операцій GP , CP й еволюції СПП. Їх сутність висвітлює

Означення 7. Об'єднанням $\cup$, перетином $\cap$ і доповненням $\neg$ множин складних об'єктів (відповідно, виходом однойменних операцій $\cup^c, \cap^c, \neg^c$ на 2^C) є результат однойменних теоретико-множинних операцій над множинами

–операндами, якщо він є конфігурацією об'єктів (відповідно, КПВ) або ж $\emptyset$ – інакше;

$$\begin{aligned} OO &= \{\cup, \cap, \neg\}; CO = \{\cup^c, \cap^c, \neg^c\}; \\ GP &= \{ap; ce; fp\} \cup OM; CP = PV \cup NV, \end{aligned} \quad (13)$$

де $ap(g)$ – автономне проектування ПП;

$ce(g; EP, Cr)$ – композиція множини наявних ПП (EP) операціями OO, CO , яка оптимізує прийнятий критерій Cr ефективності розроблення ПП (тривалість, вартість, окупність, якість тощо);

OM – операції зміни певного ПП з СПП, які оптимізують критерій Cr (тобто оптимальним чином застосовані операції з множини CP);

fp – довільна композиція ($^\circ$) операцій з множини OM та операцій ap і ce ;

PV і NV – операції передбаченої й непередбаченої зміни певного ПП з СПП.

У виразі (13)

$$\begin{aligned} ap(g) &= \langle ip; g; \tau(g) \rangle, g = \cup_{o \in g} \{o\}, \tau(g) = \cup^c \{ \tau(o), o \in g \}; \\ ce(g; EP) &= \langle ip; g; \tau(g) \rangle, g = \overset{\circ}{\cup}_{ep \in EP} \varepsilon_i(\eta_i(g(ep))), \\ \tau(g) &= \overset{\circ}{\cup}_{ep \in EP} \varepsilon_i(\eta_i(\tau(g(ep)))), \end{aligned} \quad (14)$$

де $\varepsilon_i \in \{\cup, \cap\}$; $\varepsilon_i^c \in \{\cup^c, \cap^c\}$; η_i, η_i^c – операції включення конфігурації об'єктів і КПВ деякого ПП у композицію, її вилучення й доповнення.

Далі в (13), множина PV включає узгоджені операції проектування близьких передбачених варіантів певного ПП pm_0 з моделлю (12) для полегшення побудови нових ПП у межах СПП і адаптації pm_0 до передбачених змін:

$$\begin{aligned} \delta^p(pm_0; ov) &= \langle ip^*; SO_0 \setminus \{ov\}; \{ \tau(o), o \in SO_0 \} \setminus \tau(ov) \rangle; \\ \alpha^p(pm_0; ov^*) &= \langle ip^*; SO_0 \cup \{ov\}; \{ \tau(o), o \in SO_0 \} \cup \tau(ov) \rangle; \\ \kappa^p &= \alpha^p \circ \delta^p, \end{aligned} \quad (15)$$

де $\delta^p, \alpha^p, \kappa^p$ – вилучення, долучення й варіантно-передбачена заміна варіанта $(ov, \tau(ov))$ в точках варіабельності властивостей і КПВ певного ПП іншим варіантом $(ov^*, \tau(ov^*))$ з відповідними змінами в конфігурації інтерфейсів pm_0 .

Нарешті, в NV об'єднано операції проектування непередбачених близьких і далеких варіантів ПП для адаптації до непередбачених змін поширенням функцій і якості поза межі СПП:

– узгоджене долучення (α^n) й заміну (κ^n) варіанта в точках варіабельності об'єктів і КПВ у pm_0 (11) новим об'єктом і КПВ, який він реалізує, із зазначенням його типу як варіанта (отримується близький варіант);

– узгоджене коректне (за означенням 5) вилучення (δ^{un}) обов'язкового КПВ $s \in \tau(g)$ й об'єкта, $o = \tau^{-1}(g)$, який він реалізує, “згори” (тобто з поповненням вихідних інтерфейсів (8) під-

порядковуючого КПВ для виконання умови (9) в разі необхідності) – формується далекий варіант;

– вилучення “знизу” ($\delta^{\text{дн}}$) – навпаки, з поповненням вхідного інтерфейсу підпорядкованого КПВ для виконання (9);

– вилучення “зсередини” ($\delta^{\text{мн}}$) із взаємним узгодженням нових “сусідніх” КПВ та змінами в конфігурації властивостей;

– узгоджене перепідпорядкування ($\chi^{\text{н}}$) об’єкта й КПВ у pm_0 – далекий варіант за використання обов’язкових об’єктів і близький – інакше;

– довільне вилучення (δ^{rc}), долучення (α^{rc}), заміну (κ^{rc}) й перепідпорядкування (χ^{rc}) тільки КПВ у їх конфігурації $\tau(g)$ з pm_0 з виконанням умови (9) – непередбачений за КПВ варіант.

Зауваження 3. Операції ap , fp , NV , пов’язані з долученням непередбачених функцій/даних ПП, індукують операцію зовнішньої компонентної алгебри [4-6] – розгортання КПВ їх реалізації в компонентному середовищі СПП [5], утвореному елементарними КПВ (листками графу (C, R_C)).

Корисні властивості об’єктно-компонентної моделі СПП

Результатом зіставлення визначень алгебраїчної системи й ізоморфізму цих систем [11] з означеннями 2, 7, виразами (5)-(8) і лемою 1 є

Теорема. Відображення τ (6) – ізоморфізм об’єктної й компонентної підмоделей СПП як алгебраїчних моделей типу $\langle 2, 2, 2, 2, 2 \rangle \langle O; R_O \rangle$ і $\langle C; R_C \rangle$ та як алгебр типу $\langle 2, 2, 1 \rangle \langle 2^O; OO \rangle$ і $\langle 2^C; CO \rangle$.

Наслідок 1. Відображення, що зіставляє об’єкту його інтерфейс (5), а цьому інтерфейсу – КПВ (8), для якого він є вхідним, – ізоморфізми алгебраїчних моделей типу 2 $(O; R_O)$, $(I; R_I)$, (C, R_C) , які перетворюють точки варіабельності/варіанти складних об’єктів у точки варіабельності й варіанти інтерфейсів і КПВ.

Наслідок 2. Образ конфігурації об’єктів g при відображенні τ (6) $\tau(g) = (SC = \{\tau(o), o \in SO\}; R_C)$ – конфігурація КПВ.

Наслідок 3. Листку $ol = (fl, d(fl))$ графу $(O; R_O)$ що зіставляє елементарній функції дані, відповідають листки графів (I, R_I) й (C, R_C) : елементарний інтерфейс $il(ol) = \langle id, M(fl), D(d(fl)) \rangle \in I$ та КПВ

$$cl = \tau(ol); II.cl = il(ol); IO.cl = \{il(ol)\}; S.cl_i \cap S.cl_j = \emptyset.$$

На відміну від програмних модулів для предметно-орієнтованих методів $M(fl)$, реалізації КПВ-вузла, що підпорядковує cl в (C, R_C) , не є предметно-орієнтованими, але забезпечують виклик коду з $S.lc$ і можуть надаватися інтегрованими середовищами збирання КПВ.

Наслідок 4. Довільна конфігурація об’єктів g для ПП з моделлю (12) реалізується підмножиною елементарних КПВ – листків графу (C, R_C) – $\tau(g)$ без дублювання методів та атрибутів даних.

Зауваження 4. Операції проектування GP не утворюють алгебри. Операції зміни ПП PV , NV формують двоосновні алгебри його варіантів:

а) передбачених – $\langle PR, V; \delta^p, \alpha^p, \kappa^p \rangle$;

б) непередбачених – $\langle NPR, MO; \delta^{\text{nt}}, \delta^{\text{dt}}, \delta^{\text{mn}} \rangle$, $\langle CPR, RA; \delta^{\text{rc}}, \alpha^{\text{rc}}, \kappa^{\text{rc}}, \chi^{\text{rc}} \rangle$.

Тут MO – обов’язкові об’єкти в pm_0 ; PR – множина моделей вигляду (12) з MO ; V – варіанти в g_0 ; NPR і CPR – моделі вигляду (12) хоча б з одним, але не всіма, об’єктами з MO (відповідно, КПВ з $\tau(MO)$); $RA \supset MO$ – відомі при проектуванні СПП моделі КПВ (7).

Перша алгебра відіграє для ПП і неелементарних КПВ ту ж роль формального засобу адаптації до змін вимог і умов використання, що й внутрішня алгебра рефакторингу в ОКМ [4-6], незастосовна до цих КПВ за відсутності доступу до їх реалізацій, – для елементарних КПВ. Друга відповідає моделі реінжинірингу КПВ [4-6]. Тому їх названо проміжними компонентними алгебрами структурного рефакторингу й реінжинірингу.

Узгоджені ОКМ-моделі СПП (1)-(8) і ПП (12) утворюють ядро інформаційного середовища ОКМ-розроблення СПП. Воно завершує конструктивну деталізацію своїх прототипів, керованих МВ та інтегрованою моделлю варіабельності [9, 10]. Зміст деталізації надає рис. 2.

Включення об’єктної й компонентної підмоделей та формованого ними середовища з елементарних КПВ до інформаційного середовища процесу інженерії домену, показане на рис. 2, підтримує уніфіковану реалізацію операцій PV і



Рис.2. Розвиток середовища побудови СПП

іNV (13) конфігуруванням цих КПВ, забезпечуючи адаптовність ПП до передбачених і непередбачених змін. Адаптовність СПП у цілому забезпечується внесенням до цього середовища діагностичної ОКМ-моделі варіабельності. Вона уможливорює операції еволюції СПП, розглянуті далі.

Підхід до використання об'єктно-компонентної моделі СПП

Пропонується створення формального апарату реалізації функцій ОКМ-моделі СПП (1)-(8) у середовищі, поданому на рис. 2:

- 1) узгодження МВ СПП, відповідних наявним ПП та очікуваним потребам їх користувачів;
- 2) підтримки проектування ПП і їх варіантів;
- 3) обґрунтованого кількісного контролю рівня варіабельності СПП;
- 4) підтримки еволюції СПП за неадекватності умовам використання.

Першій функції зіставлені три засоби:

- 1) новітні техніки відновлення МВ (з коду ПП, конфігураційних файлів, специфікацій) [3];
- 2) багатотуровий Процес конструктивного аналізу ПрО [12], розроблений за участі автора з використанням Інженерії співпраці для експертного формування матриці приналежності властивостей ПрО потенційним ПП з СПП;
- 3) постановка задачі про ранець для раціонального добору елементарних КПВ у СА:

$$\sum_{i=1, \dots, n} \rho_i x_i \rightarrow \max, \quad \sum_{i=1, \dots, n} \lambda_i x_i \leq 0, \\ x_i = 1 \text{ при } cl_i \in C, \quad x_i \neq 1 \text{ при } cl_i \notin C,$$

де cl_i , $i=1, \dots, n$ – потенційні елементарні КПВ, визначені у Процесі аналізу; ρ_i – сукупна оцінка корисності cl_i, ol_i для СПП; λ_i – витрати на внесення cl_i до компонентного середовища СПП.

Функції й дані з МВ, відновленої за наявними ПП, утворюють інформаційний контекст турів Процесу аналізу, що мають деталізувати матрицю 2) до об'єктної підмоделі ОА з (1). Компонентна підмодель СА= τ (ОА) і середовище з її елементарних КПВ забезпечують адаптовність як наявних, так і нових ПП завдяки операціям CP (13) їх структурного рефакторингу й реінжинірингу та рефакторингу КПВ [5].

Для другої функції пропонується розвиток операції об'єднання КПВ в ОКМ [7,8], а саме операція збирання конфігурації КПВ. Її результат – КПВ, званий збіркою. Вхідний інтерфейс збірки є покоординатним, у сенсі леми 1, об'єднанням вхідних інтерфейсів елементарних

КПВ з конфігурації та методів mch^t , $t=1, \dots, 4$ вибору між ними для всіх КПВ-точок варіабельності з елементарними варіантами типу t (у сенсі зауваження 1). Множини вихідних інтерфейсів і реалізацій збірки – об'єднання однойменних множин її елементів. Необхідна також постановка оптимізаційної задачі вибору схеми збирання, оптимальної за певним критерієм (якість, окупність, вартість тощо).

Складниками формального апарату для третьої функції є діагностична ОКМ-модель варіабельності та операції її контролю [11]. Модель розвиває первинну нечіткісну модель А.Мецгера типізацією проявів її неадекватності потребам у ній. Це узгоджений з ОСМ (1) кортеж

$$EV = \langle il; \langle vl_i; rl_i, pl_i, ml_i, t \in T \rangle; ia; \langle va_i; ra_i, pa_i, pu_i, t \in T \rangle; \langle NA_{ti}, t \in \{o, i, a\}; i=1, \dots, 5 \rangle \rangle, \quad (16)$$

де il, ia – інтегральні рівні автономної варіабельності СПП та її відповідності потребам користувачів і розробників ПП;

vl_i і va_i – проміжні рівні автономної варіабельності й адекватності, формовані артефактами типу $t \in T = \{f, d, o, i, a\}$ – об'єктами функцій (f), даних (d), складними об'єктами (o), інтерфейсами (i), КПВ (a);

rl_i, pl_i і ml_i – вкладені рівні варіабельності, передбаченої й реалізованої для артефактів типу t , а також рівня відповідності між ними;

ra_i, pa_i, pu_i – вкладені рівні адекватності передбаченої й реалізованої варіабельності артефактів типу t потребам користувачів і розробників ПП, а також ефективності використання КПВ;

NA_{ti} – артефакти типу t , що обумовлюють виявлену неадекватність варіабельності типу i .

Аналіз інформаційних середовищ побудови СПП [1-3,9] висвітлює такі типи неадекватності:

- 1) “надлишковість” – наявність артефактів, невикористованих/неокупних при побудові СПП;
- 2) “неповнота” – відсутність функцій/даних чи КПВ для них, запитаних значущою більшістю користувачів і розробників ПП;
- 3) “клони” – наявність артефактів, повністю взаємозамінних під час побудови СПП;
- 4) “хибні обмеження” – артефакти, яким потрібна зміна статусу як точки варіабельності;
- 5) “хибні залежності” – артефакти, що потребують перепідпорядкування і/або зміни типу як варіанти певної точки варіабельності.

Листки, вузли та інтегральні рівні моделі EV (16) доповнюються експертно формованими вербально-числовими шкалами – наборами пар “опис стану СПП; рівень критичності”.

Останній функції ОСМ відповідають три групи операцій над елементами NA_i (16), що розвивають операції [10] за допомогою ОКМ:

1) автономна еволюція або заміна КПВ з середовища без зміни інтерфейсів (внутрішня ОКМ-алгебра рефакторингу елементарних КПВ і алгебра структурного рефакторингу складних КПВ чи зовнішня ОКМ-алгебра для їх заміни);

2) долучення/вилучення функцій і даних та актуалізація їх відношень через узгоджену актуалізацію графів об'єктів і КПВ без зміни листків;

3) узгоджена еволюція довільних КПВ з компонентної підмоделі СПП із змінами їх інтерфейсів і зв'язків у графі (C, R_C) в (1) та актуалізація графу об'єктів (ОКМ-модель реінжинірингу елементарних і алгебра структурного реінжинірингу – складних КПВ, зовнішня ОКМ-алгебра).

Висновки

Вперше в інженерії СПП критичне обмеження моделі властивостей – відсутність зв'язку з простором рішень – опрацьовано за допомогою об'єктно-компонентного методу К.М.Лавріщевої-В.М.Грищенка. Її перетворено в об'єктну підмодель СПП: граф об'єктів ПрО, інтерфейси яких подають спільні й змінні функції та дані ПП з їх залежностями, та операції добору функцій для ПП. Цій підмоделі зіставлено компонентну – ізоморфний граф КПВ реалізації інтерфейсів з операціями їх конфігурування. Долучено операції проектування та алгебри структурного рефакторингу й реінжинірингу ПП з операціями їх зміни, передбаченої й непередбаченої в СПП.

Для забезпечення адаптовності ПП й СПП надано нові похідні моделі: ПП та діагностичну модель варіабельності. Перша подає центри передбачених змін ПП узгодженими точками варіабельності об'єктів і КПВ, а межі – їх варіантами. Друга є засобом визначення моменту й змісту еволюції СПП та контролю її наслідків.

Описано склад формального апарату застосування пропонуваного об'єктно-компонентних моделей для забезпечення адаптовності ПП з СПП.

Ці моделі й апарат засвідчують перспективність розвитку методу К.М.Лавріщевої-В.М.Грищенка для створення об'єктно-компонентної технології побудови адаптовних СПП. Їх конкретизація для платформи MS.Net й апробація в інструментально-технологічному комплексі ІПС НАНУ – предмет подальших досліджень автора.

Список використаних джерел

1. V. d. Linden F., Schmid K., Rommes E. Software product lines in action: the best industrial practice in product line engineering. – Heidelberg: Springer, 2007. – 340 p.
2. Lee K. et al. Concepts and Guidelines of Feature Modeling for Product Line Software Engineering. // 7th Int. Conf. on Software Reuse: proceedings. – Springer-Verlag, 2002. – P.62–77.
3. She S. et al. Reverse Engineering Feature Models // ICSE '11: international conference, May 21–28, 2011, Waikiki, Honolulu, USA: proceedings. – 2011. – P. 128-139.
4. Lavrisheva K. et al. New Theoretical Foundations of Production Methods of Software Systems in Generative Programming Context. Electronic monograph, in: UK-2011, vol. 67. – Kiev, 2011. – (in Ukrainian).
5. Lavrisheva E., Grischenko V. Assembly Programming. Basics of Software Industry. – Kiev: Naukova Dumka, 2009 (2nd ed.). – 372 p. (in Russian).
6. Grischenko V. The Method for Object-Component Software Design // Problems in Programming – 2007. – N 2. – P. 113-125. (in Russian).
7. Lavrisheva E. Programming methods. Theory, Engineering, Practice. – Kiev: Naukova Dumka, 2006. – 451 p. (in Russian).
8. Lavrisheva E. et al. Object-Component Design. Theoretical and Applied Issues // Visn., Ser. Fiz.-Mat. Nayky, Kyiv Univ. im. Tarasa Shevchenka. – 2013. – Special Issue. (in Ukrainian).
9. Lavrisheva K. et al. The Theoretical View for Software Family Variability Management // Visn., Ser. Fiz.-Mat. Nayky, Kyiv Univ. im. Tarasa Shevchenka. – 2011. – N 1. – P. 45-53. (in Ukrainian).
10. Slabospitskaya O. An approach for Software Family sound evolution // TAAPSD'2012: 9 int. conf., December 3-7, 2012, Kyiv, Ukraine: proceedings. – Kyiv, 2012. – P. 274-278. (in Ukrainian).
11. Artamonov V., Saliy V., Scorniyakov L. General algebra. – Moscow: Nauka, 1991. – 348 p. (in Russian).
12. Koval G. et al. Software Family Engineering Enhancement with Agile Methodologies // Problems in Programming – 2010. – N 2-3. – P. 261-270. (in Ukrainian).

Надійшла до редколегії 15.07.13

УДК 004.891.3: 519.7

Сугоняк І.І., к.т.н.¹

СППР діагностики захворювань з використанням систем нечіткого виводу

В роботі побудовано алгоритми функціонування СППР з діагностики різних типів захворювань на основі системи нечіткого виводу Мамдані з використанням методу аналізу ієрархій.

Ключові слова: СППР, нечітке моделювання, бази знань, діагностування злоякісних пухлин.

¹Житомирський державний технологічний університет, 10000, Житомир, вул. Черняхівського 103
e-mail: isygon@mail.ru

Inna I. Sugonyak, PhD¹

DSS diagnosis of diseases using fuzzy logic system

The article presents a result of development of algorithm for DSS system in medical practice, which provide complex analysis of existing medical practice for certain types of diseases.

Key words: DSS, fuzzy modeling, knowledge base, diagnosing malignant tumors.

¹Zhytomyr State technological university, Zhytomyr, Chernyahovskogo str, 103
e-mail: isygon@mail.ru

Статтю представив д.т.н., проф. Акіменко В.В.

Медичне рішення відноситься до слабо структурованих галузей знань, побудова алгоритмів підтримки прийняття рішень в яких є складним процесом [1]. Для підвищення ступеня обґрунтованості рішень щодо доцільності додаткових обстежень і всебічного аналізу симптомів онкохвороб на ранніх стадіях, вибору ефективних в конкретному випадку схем лікування на основі накопиченого досвіду та лікарської практики можливим є використання СППР [2,3].

Конкретні прояви злоякісних новоутворень залежать від багатьох факторів. Типові ознаки об'єднуються у дві групи - загальних і місцевих симптомів. Залежно від локалізації та перебігу хвороби, першими можуть з'явитися як загальні, так і місцеві ознаки. В такому випадку можливим є використання нечітких алгоритмів виводу Мамдані [4], де рішення може бути прийняте на основі попереднього досвіду експертів та системи логічних узагальнень.

Нехай стан j -го пацієнта визначається масивом наявних симптомів $E_j^c = \{e_i^j\}$. В свою чергу k -те захворювання визначається на множині з n симптомів, що згруповані у m груп за вибраною множиною ознак і утворюють ієрархічну множину (рис.1). Такий поділ полягає у групуванні за наступною ієрархією:

Кожному симптому відповідає лінгвістична оцінка за шкалою 0-1 (див. приклад табл. 1) та

визначений за методом експертних оцінок або статистичних досліджень ваговий коефіцієнт ймовірності прояву симптому. Групи симптомів для окремої хвороби також ранжуються за важливістю із використанням методу попарних порівнянь.

Таблиця 1

Простато-специфічний антиген (PSA)

Діапазон значень	Оцінка	Бали
Менше 4 нг/мл	Низький	0
4 -до 10 нг/мл	Проміжний	0,5
Більше 10 нг/мл	Підвищений	0,75

Стан i -го пацієнта характеризується множиною симптомів що відповідають захворюванням F_{ik} , та наявними в нього можливостями щодо лікування – вектором протипоказань $P_i = \{p_{ij}\}$ - де $p_{ij} \in [1..5]$ оцінка рівня граничного впливу на j -ту систему органів, та наявним рівнем ресурсів R_i - сумарний час тривалості процедур за один день. Лікар за результатами огляду пацієнту складає матрицю попарного порівняння складності порушень для їх ранжування за важливістю(черговістю) втручання. За даними матриці визначаються ранги λ_{ik} кожного з порушень з метою визначення пріоритетності лікування. Для множини напрямків впливу на організм

$Q = \{Q_r\}$ поставлені у відповідність методи лікування $L = \{l_s\}$. Кожен з методів визначений у просторі властивостей $l_s = \langle E, R_s, T, T_y \rangle$, де E – рівень впливу методу (за лінгвістичною оцінкою низький, помірний, сильний), R_s – тривалість процедури, T – тривалість застосування методу, T_y – (періодичність застосування методу), причому $E = f(T, T_y)$. Тоді оптимальним лікуванням є така матриця методів лікування, що має наступну структуру: $M = \langle L_{opt}, t_1, T, T_y \rangle$, де L_{opt} – підмножина методів лікування що визначена для i -го пацієнта, t_1 – день початку застосування методу (1 вважається день початку лікування).

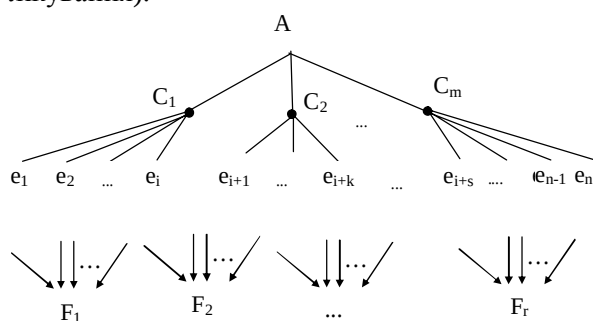


Рис.1. Загальна структура ієрархії симптомів для визначення захворювання

Необхідним є розробка методів підтримки прийняття рішень, для вибору на множині існуючих захворювань одного або сукупності, таку що пояснює всі наявні у пацієнта симптоми; визначення оптимального комплексу методів лікування та планування їх у часі із врахування викладених вище обмежень.

Кожному захворюванню відповідає нечітка множина симптомів, що визначається наступним чином:

$E_i = \{\mu_{ik} / e_{ik}\}$. μ_{ik} – визначає нечітку ймовірність прояву даного симптому у хворих, e_{ik} .

Для визначення ймовірності прояву симптому при даному захворюванні даного показника розглянемо два випадки:

а) симптом є бінарно визначеним (тобто або він проявляється або ні). В такому випадку для визначення його ваги використовується класичне визначення ймовірності:

$$\mu_i = \frac{N_i}{N}, \text{ для опрацювання бази даних}$$

діагностованих хворих,

або якщо визначається за експертним оцінюванням то

$$\mu_{ik} = \sum_{i=1}^n p_i v_i / v_i, \text{ де } v_i - \text{ваговий}$$

коефіцієнт що відображує ранг експерта.

б) якщо симптом не є бінарним, тобто можливою є наявність якісно різних проявів одного симптому, його ймовірність розраховується за спрощеною формулою Байеса:

$\mu_{ik} = p(N_{ir} | N_i) * p(N_i)$, де $p(N_{ir} | N_i)$ – ймовірність прояву симптому із g -статусом, серед всіх випадків прояву симптому, $p(N_i)$ – загальна ймовірність прояву симптому у хворого даним захворюванням, далі можливим є використання декартового добутку станів системи, що забезпечить наявність декількох альтернативних нечітких множин симптомів захворювання або використання співвідношення $\mu_{ik} = \max\{\mu_{ik}(N_{ir})\}$ забезпечить один варіант опису. При формуванні нечіткої множини симптомів захворювання для порівняння із станом хворого до неї включається тільки одне значення, що якісно відповідає симптому хворого.

Уточнення коефіцієнту μ_{ik} в процесі роботи системи відбувається шляхом врахування результатів поточної статистичної оцінки даних бази пацієнтів як думки окремого експерту, ранг якого підвищується із збільшенням кількості даних.

Інтегральна оцінка захворювання визначається за методом Сааті:

$$F_i = ME_i, (1)$$

де $M\{\pi(AC_j)\mu_{ik}(C_j e_k)\}$ – вектор вагових коефіцієнтів дуг останнього рівня.

Інтегральна оцінка наявних у пацієнта симптомів визначається аналогічним чином:

$$F_i^c = M^* E^c, \text{ де } M^* - \text{розширений вектор}$$

вагових оцінок дуг останнього рівня ієрархії симптомів захворювання, в якій за умови наявності в пацієнта симптомів, що не належать хворобі приймаються нульові значення для коефіцієнтів.

Для розрахунку допустимого значення $\mathcal{E}$ – від множини еталонних захворювань необхідним є: розрахунок граничних інтегральних оцінок захворювання у випадку наявності для кожного захворювання однієї множини симптомів, або мінімальної та максимальної інтегральної оцінки у випадку наявності альтернативних множин симптомів.

Граничними інтегральними оцінками захворювання у випадку наявності для кожного захворювання однієї множини симптомів вважається оцінка співвідношення (1), та оцінка, що розраховується за співвідношенням (1) лише для підмножини симптомів $\mu_{ik}(C_j e_k) \geq \mu^*$ (для достатньої достовірності приймається $\mu^* = 25\%$). В обох випадках ε визначається як:

$$\varepsilon = (F_{\max} + F_{\min}) / 2$$

На етапі діагностування система може отримати наступні результати:

1) Пацієнт має одне захворювання у випадку якщо: нечітка множина симптомів відповідає нечіткій множині ознак хвороби, $E^c = E_i$, або $E^c \in E_i$. Для порівняння множини симптомів E_j^c присвоюються значення μ_{ik} що розраховані для хвороби. Якщо інтегральна оцінка множини симптомів має відхилення, що не перевищує певне значення ε від еталонної множини симптомів, що наявні в базі знань: $|F_i^c - F_i^{ek}| \leq \varepsilon$.

2) Пацієнт має декілька захворювання у випадку якщо: нечітка множина симптомів перетинається з нечіткою множиною ознак хвороби, $E^c \cap E$, у такому випадку проводиться пошук сукупності з k хвороб що задовольняють умові: $E^c \in \bigcup_{i=1}^k E_i$ для кожної хвороби розраховується

інтегральна оцінка підмножини симптомів E_k^c , що наявні у пацієнта і відповідають певній хворобі з визначеної сукупності і, відхилення від еталонної множини симптомів, що наявні в базі знань окремих захворювань в такому випадку не перевищує певне значення ε : $|F_i^c - F_i^{ek}| \leq \varepsilon$, для всіх хвороб.

3) Пацієнт потребує додаткове обстеження у випадках, якщо:

- $E^c \notin E_i$, або $E^c \subseteq \sum_{i=1}^k E_i$ та немає можливості змінити масив симптомів, виключивши незначні з його точки зору, додавши інші. Обстеження

визначається у напрямку, що відповідає масиву захворювань $\inf\{E^c - E_i\}$.

- $E^c = E_i$, або $E^c = \sum_{i=1}^k E_i$, але $|F_i^c - F_i^{ek}| > \varepsilon$.

в такому випадку крім обстеження можна призначити лікування, так як відхилення може бути обумовлене індивідуальною реакцією пацієнта на захворювання.

4) знайдено декілька взаємовиключних захворювань, або декілька можливих сукупностей захворювань -- тоді найбільш ймовірним є захворювання з найменшим ε , за необхідності лікар може направити пацієнта на додаткове обстеження вже для визначеного переліку захворювань.

Загальна структура процесу прийняття рішення щодо вибору та планування оптимального комплексу лікувальних заходів наведена на рис. 2

Метою блоку S1 є визначення напрямів лікування, що відповідають визначеним порушенням стану здоров'я пацієнта за умови постійного оновлення матриці знань. Результатом моделювання в блоці S1 є визначення:

$$\bigcup_r \left[\bigcap_k (F_k = 1) \right] \rightarrow Q_{kr} = Q_k, k = \overline{1, m}, r = \overline{1, n}, (1)$$

тобто набору правил вигляду ЯКЩО $(F_{k_i} = 1) \text{ І } (F_{k_j} = 1) \text{ АБО } (F_{k_s} = 1) \text{ ТО } y = d_m$.

Матриця знань формується за визначеними симптомами та існуючими методами лікування доступними в даний час медичному закладу або пацієнту (під час роботи проводиться до навчання системи). Фрагмент початкового варіанту матриці знань та правил вибору напрямків лікування наведений на рис.3.

Основною метою блоку S2 – є формування правил визначення методів лікування, що відповідають визначеним напрямам Q_r та є доступними для застосування в даний час. База знань для другого блоку також є продукційною і визначається набором правил вибору методів, аналогічних (1).

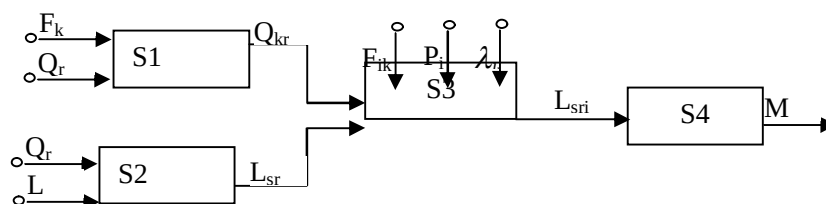


Рис.2. Структура моделі прийняття рішень щодо визначення матриці методів лікування пацієнта

BTA (F ₁)	NMP22 (F ₂)	CA15-3 (F ₃)	CEA (F ₄)	...	Q _t
2	3	4	5	...	m+1
1	0	0	1	...	Q1 - Рак сечового міхура, УЗД
0	0	1	0	...	Q2- Рак яєчників, цистоскопія
0	0	0	0	...	Q1 - Меланома
...	...	...	...	...	...

$$\bigcap_{i=1,3} F_i \bigcup F_i \rightarrow Q_{kr} = Q_1;$$

$$\bigcap_{i=1,3} F_i \bigcup F_i \rightarrow Q_{kr} = Q_2$$

$$F_5 \rightarrow Q_{kr} = Q_3...$$

Рис.3. Фрагмент матриці знань, набору правил системи та прототип інтерфейсу системи

Пацієнт: 12431

Симптом	Значення
BTA	0
NMP22	0
CA15-3	1
CEA	0

Оцінка

Попередній діагноз - рак яєчників
Оцінка достовірності - 0,83%
Коефіцієнт узгодженості експертів - 0,9
Додаткові рекомендації - цистоскопія

В блоці S3 – виконується вибір напрямків лікування та їх впорядкування за пріоритетами. Пріоритетність напрямку залежить від його рівня $\lambda_{ik}^* = \sum_j \lambda_{ikj} / \sum_k \sum_j \lambda_{ikj}$, λ_{ikj} - оцінка попарного порівняння симптомів, що спостерігаються у і-го пацієнта. Такий пріоритет присвоюється і методам лікування, що підходять для даного напрямку. Необхідний рівень впливу метода лікування ставиться у відповідність до рівня порушення у пацієнта. Утворена підмножина методів $L_{si} = \langle E, R, P, T, T_y \rangle$, де $P = \{P_{ik}\}$ - протипоказання застосування даного методу – складає множину допустимих лікувальних заходів пацієнта, R – тривалість однієї процедури. Тоді множина можливих лікувальних заходів $L_{sri} = \sigma_{P_{ik} \neq P_i}(L_{si})$, де σ - операція вибірки. Тобто до підмножини L_{sri} входять лише ті методи, які не мають протипоказань, що співпадають з вектором протипоказань пацієнта.

Метою блоку S4 є вибір оптимального плану лікування. Задачею оптимізації в даному випадку є визначення Парето-оптимальної множини методів, що забезпечують наступні вимоги: максимізацію лікувального ефекту:

$$F_k(x) = \max(E x_j^k), \quad (2)$$

та мінімізацію часу витраченого на лікування:

$$T_k(x) = \min(T * x_j^k / T_y) \quad (3)$$

де x_j^k - метод лікування, що належить до k-го напрямку лікування.

Для системи притаманні наступні обмеженнями: набір правил що визначає медичну можливість застосування даних методів

одночасно $\bigcup_i \bigcap_j x_j^i = 1$; метод може бути

обраним або не обраним $x_j^i \in [0;1]$; тривалість процедур в день не перевищує обмежень для пацієнта за часом:

$$\sum_k \sum_j T_j^k * x_j^k / T_{y_j}^k / \max\{T_j^k\} \leq R_i \quad (4)$$

Розв'язання задачі багатокритеріальної оптимізації (2-4) доцільно виконати лексикографічним методом.

Висновки.

В роботі запропоновані алгоритми прийняття рішень щодо ранньої діагностики онкозахворювань та вибору комплексу методів лікування пацієнтів. Розроблена СППР функціонує в трьох режимах: навчання - початкове наповнення бази знань та формування правил, підтримки прийняття рішень – визначення плану лікування та його корекція при зміні стану пацієнта; донавчання – внесення нових знань в систему за результатами досліджень та досвіду лікування пацієнтів.

Список використаних джерел

1. Moiseev N.N. Mathematical problems of system analysis. Moscow: Nauka. - 1986. - 234 p. (in Russian)
2. Akimenko V.V. The decision support system system under conditions of mixed information for the environmental monitoring of the atmosphere // Cybernetics and Systems analiz.-2000. - № 5.- p.151-167. (in Russian)
3. 6. Voloshin O.F., Mashchenko S.O. Models and methods of decisions making. K.: KNU im.T.Shevchenko. - 2010. - 336. (in Ukraine)

Надійшла до редколегії 14.11.2013

УДК 004.93

Тарануха В.Ю., асистент

Застосування класів основаних на канонічних формах слів та на граматичних класах в задачі редукції n-грамної моделі мови для розпізнавання української мови

Стаття присвячена дослідженню n-грамної моделі української мови для розпізнавання та методу редукції, призначеному для зменшення розміру моделі з якомога меншою втратою якості розпізнавання

Ключові слова: n-грамна модель мови, редукція моделі, модель на класах.

Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Глушкова 4д, e-mail: taranukha@mail.ru

V.Y. Taranukha, assistant

Applying classes based on canonical forms of words and the grammar classes in the problem of reduction of n-gram language model for recognition of the Ukrainian language

The article investigates n-gram language model for Ukrainian language recognition and method of reduction for the model aimed at least recognition rate loss.

Key words: n-gram language model, model reduction, class-based model.

Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova st., 4d, e-mail: taranukha@mail.ru

Статтю представив чл.-кор. НАН України, д.ф.-м.н., проф. Анісімов А.В.

Необхідність розробки та оптимізації моделей для розпізнавання мови зумовлена збільшенням обсягу даних, що потребують переведення в текстову форму. Однією з найбільш ефективних моделей природної мови є статистична модель на основі n-грам[1], яка оцінює ймовірність появи ланцюжка з n слів у деякому тексті. Така модель створюється в результаті статистичного аналізу навчального текстового корпусу.

Побудова та використання моделей

Послідовність слів мови $w_1...w_n$ називається n-грамою довжини n . Модель на основі n-грам передбачає, що текст або мовлення будуть мати характеристики, що дозволяють їх описати за допомогою марківської моделі. Тобто, ймовірність послідовності слів можна буде оцінити явно. Для послідовності слів $w_1...w_i$ маємо

$p(w_1...w_i) = p(w_i | w_1...w_{i-1})p(w_{i-1} | w_1...w_{i-2})...p(w_1)$ (1)
Оцінка ймовірності базується на частотах відповідних n-грам

$$\hat{p}(w_i | w_{i-n+1}...w_{i-1}) = \frac{C(w_{i-n+1}...w_{i-1}w_i)}{C(w_{i-n+1}...w_{i-1})} \quad (2)$$

де $C(w_{i-n+1}...w_{i-1}w_i)$ - частота відповідної n-грами.

Така оцінка є оцінкою максимальної

© В.Ю. Тарануха, 2013

правдоподібності. Недоліком оцінки (2) є те, що в реальних корпусах не представлені всі можливі n-грами. Виникає потреба в застосуванні методу для згладжування частот та ймовірностей відповідних n-грам.

Для цього використовуються різні методи [1], в даній статті розглядаються два з них. Згладжування з поверненням Віттена-Бела та згладжування з поверненням Катца.

$$\hat{p}(w_i | w_{i-n+1}...w_{i-1}) = \begin{cases} d(w_{i-n+1}...w_i), C(w_{i-n+1}...w_i) > 0 \\ \alpha_{w_{i-n+1}...w_{i-1}} \hat{p}(w_i | w_{i-n+2}...w_{i-1}) \text{ інакше} \end{cases} \quad (3)$$

де $d(w_{i-n+1}...w_i)$ - відповідним чином згладжене значення $C(w_{i-n+1}...w_{i-1}w_i)$, $\alpha_{w_{i-n+1}...w_{i-1}}$ - відповідний коефіцієнт, що визначає ймовірнісну масу, перерозподілену для побудови ймовірностей на n-грамах моделі меншого порядку.

$$\alpha_{w_{i-n+1}...w_{i-1}} = \frac{\beta_{w_{i-n+1}...w_{i-1}}}{\sum_{\{w_i: C(w_{i-n+1}...w_i)=0\}} \hat{p}(w_i | w_{i-n+2}...w_{i-1})}, \quad (4)$$

$$\beta_{w_{i-n+1} \dots w_{i-1}} = 1 - \sum_{\{w_i: C(w_{i-n+1} \dots w_{i-1} w_i) > 0\}} d(w_{i-n+1} \dots w_{i-1} w_i). \quad (5)$$

Для методу Віттена-Бела параметр d оцінюється так:

$$d_{WB}(w_i | w_{i-n+1} \dots w_{i-1}) = \frac{C(w_{i-n+1} \dots w_{i-1} w_i)}{C(w_{i-n+1} \dots w_{i-1} w_i) + T(w_{i-n+1} \dots w_{i-1} w_i)}, \quad (6)$$

де $T(w_{i-n+1} \dots w_{i-1} w_i)$ - кількість типів n -грам, що передують слову w_i .

Для методу Катца, параметр d оцінюється так:

$$c^* = \frac{(c+1) \frac{N_{c+1}}{N_c} - c \frac{(k+1)N_{k+1}}{N_k}}{1 - \frac{(k+1)N_{k+1}}{N_k}}, 0 < c \leq k, \quad (7)$$

$$d_{Katz} = \frac{c^*}{c}, \quad (8)$$

де N_m - кількість n -грам, що трапились m разів, c - відповідна частота n -грами, k - параметр, зазвичай рівний 5.

Формула для c^* є адаптацією формули Гудатюринга[2].

Побудова та оцінювання моделей для української та інших слов'янських мов

Найбільшу проблему для української мови складає комплексна морфологична структура слів мови в поєднанні з нефіксованим порядком слів у реченні. Якщо на одне слово англійської мови припадає 1,7 словоформи, то в корпусі застосованому для роботи від 5,5 до 19,9 залежно від вибраного словника. Таким чином, при побудові таблиці n -грам, при $n=2$, розмір зростає принаймні в 10,47 рази, а при $n=3$ - в понад 33 рази. Крім розростання таблиці це призводить до того, що значна кількість n -грам трапляються лише 1 раз, хоча могли би мати вищу частоту. Для таких n -грам важко визначити, чи то є корисна n -грама, чи то є шум. Наприклад, метод Катца передбачає, що після обчислення параметрів моделі n -грами з частотою 1 можна відкинути.

Для оцінки якості моделі без необхідності виконувати повний експеримент з розпізнаванням використовується ентропія(9). Це дозволяє оцінити якість марківського ланцюжка, проте ігноруються певні аспекти реального розпізнавання.

$$H(X) = - \sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \quad (9)$$

Наприклад, ігнорується схожість звучання слів, що може призвести до погіршення результатів у реальному експерименті.

Крім того, має сенс обчислювати ентропію з розрахунку на слово, в потенційно нескінченному реченні, що описує мову.

$$H(L) = \lim_{n \rightarrow \infty} - \frac{1}{n} \log p(w_1 w_2 \dots w_n) \quad (10)$$

Для порівняння двох моделей зручно використовувати крос-ентропію. Нехай

$m(w_1 w_2 \dots w_n)$ - модель для імовірності $p(w_1 w_2 \dots w_n)$ тоді крос-ентропія з розрахунку на слово виражається так:

$$H(p, m) = \lim_{n \rightarrow \infty} - \frac{1}{n} \log m(w_1 w_2 \dots w_n) \quad (11)$$

Для неї відомо, що

$$H(p) \leq H(p, m) \quad (12)$$

Як видно з формули (9), що більше буде n -грам з неправильною частотою, то гірша буде оцінка моделі. Відповідно, перплексія з розрахунку на слово обчислюється так:

$$PPW = 2^{H(p, m)} \quad (13)$$

Покращення моделей

Пропонувалися різні підходи до вирішення цієї проблеми розростання таблиць та зростання ентропії для слов'янських мов, а отже і для української.

Перехід до n -грам з вільним порядком слів[3].

$$\hat{p}(w_i | w_{i-n} \dots w_{i-1}) = p(w_i | \{w_{i-n} \dots w_{i-1}\}) \quad (14)$$

Фігурні дужки позначають, що слова крім останнього збираються з довільним порядком. Чисельний експеримент показав непридатність цієї моделі для розпізнавання, через високу ентропію[3].

Використання додаткових джерел інформації, а саме - даних синтаксичного аналізатора[4]. В комплект n -грам додаються n -грами, що отримані як коректні словосполучення в синтаксичному дереві розбору. Отримано гарантований ефект підвищення якості розпізнавання. Нажаль, цей метод передбачає необхідність використання хорошого синтаксичного аналізатора, що не завжди є можливим.

Використання декомпозиції моделі на дві: модель засновану на граматичних класах, та модель засновану на канонічних формах слів. Згідно наведених оцінок для перплексії[3], при використанні обох часткових моделей одночасно

результуюча перплексія набагато більша, ніж перплексія моделі створеної лише на словоформах. Це позбавляє сенсу безпосереднє використання двох моделей, хоча відповідно до запевнень авторів[3] лишає можливість винести в модель на граматичних класах частину інформації, необхідної для боротьби з акустичною схожістю різних форм одного слова.

Використання оптимізації ентропії або перплексії моделі шляхом оцінки та вилучення шумів з моделі[5]. При цьому можна оцінювати як модель зібрану на словоформах, так і модель зібрану на канонічних формах слів. Метод показав дієвість такої оптимізації, проте він має відчутний недолік, оскільки фактично спирається в оптимізації на п-грами малої частоти, а отже, на шуми та викиди. Це не гарантує якості при тестах на інших корпусах.

Редукція на базі двох моделей

Пропонується скомбінувати підходи запропоновані в роботах[3,5]. Провести декомпозицію словоформної моделі, на дві моделі на класах. Перша група класів утворюється множиною канонічних форм слів. Друга група класів утворюється множиною граматичних класів слів. Оцінити частоти п-грам на моделі канонічних форм та моделі граматичних класів, після чого відкинути ті п-грами, що мають надто малі частоти підтвердження на обох часткових моделях.

Експерименти було проведено на п-грамах розмірності 3 і менше (триграмах, біграмах та уніграмах), зібраних з текстів стенограм Верховної Ради України. Було сформовано корпус обсягом 112,5 МБ. Для цього відповідні стенограми було зібрано з сайту <http://rada.gov.ua/meeting/stenogr>.

На корпусі було виділено словник системи з 10.000 словоформ, всі інші слова були замінені на стоп-слово “#”. Словник було пропущено через систему морфолексичного аналізу, і сформовано словники канонічних форм, обсягом 6044 одиниці, та словник граматичних класів обсягом 3770 одиниць.

Відносно великий розмір отриманих часткових словників зумовлений тим, що значну частину словника системи складають слова, що мають мало словоформ серед 10.000 а також тим, що деякі слова було вирішено не замінювати при побудові канонічних форм та граматичних класів.

Було проведено ряд експериментів з підбору ознак, за якими можна було б розділяти п-грами з низькою частотою, на дві множини: ті що варто

лишити і ті, що краще викинути.

В процесі експериментів, виявилось, що для кращих результатів варто оцінювати триграми разом з біграмами. Відповідно, або викидати, або лишати в моделі одночасно.

Для побудови моделі використовувалися 75% від корпусу, для обчислення ентропії(11) та перплексії(13) використовувалась решта – 25%.

Було проведено такі експерименти:

- 1) Без вилучення п-грам.
- 2) Вилучення всіх триграм та біграм, для яких $C(w_{i-n+1}...w_{i-1}w_i)=1$.
- 3) Вилучення поганих низькочастотних п-грам. Лишаються лише найкращі триграми, а саме: ті, що мають $C(w_{i-n+1}...w_{i-1}w_i)>1$, або $C(w_{i-n+1}...w_{i-1}w_i)=1$ але відповідні їм триграми в моделях канонічних форм та граматичних класів мають частоту не менше 3. Аналогічно для біграм.
- 4) Вилучення кращих з низькочастотних, лишаються лише триграми, що мають $C(w_{i-n+1}...w_{i-1}w_i)=1$ і відповідні їм триграми в моделях канонічних форм та граматичних класів мають частоту рівну 1. Або триграми з $C(w_{i-n+1}...w_{i-1}w_i)>1$.

Для експериментів 3 та 4 було виконано узгодження моделей за розміром, щоб вони мали однакову кількість доданих триграм та біграм, порівняно з експериментом 2 в цій серії.

Для методу Віттена-Бела отримано значення наведені у Таблиці 1.

Таблиця 1.

Експеримент для методу Віттена-Бела.

	Ентропія	Перплексія
Експеримент 1	7,225	149,6
Експеримент 2	7,887	236,7
Експеримент 3	7,77	218,3
Експеримент 4	7,94	245,6

Як видно з таблиці, якщо викинути всі випадки з частотою 1, то результат суттєво погіршується. Справді, значна кількість з них повинна трапитися, проте метод Віттена-Бела погано відновлює значення частоти для таких випадків. Добре помітно, що включення незначної кількості триграм та біграм в модель покращує результат, якщо вибір цих п-грам правильний. Цікаво виглядає результат експерименту 4, в якому отримано найгірше значення ентропії і, відповідно, перплексії. Це говорить про те, що в модель було внесено шум, тобто п-грами яким

дали частоту 1. Навіть після згладжування значення частоти цих n -грам надто велике.

Для методу Катца отримано дані, що наведені у Таблиці 2. Одразу необхідно зауважити, що метод Катца, що спирається на формулу Гуда-Тьюринга очікувано працює краще, аніж метод Віттена-Бела, що було показано в [1]. Як видно з Таблиці 2, якщо викинути всі випадки з частотою 1, то результат також погіршується, як і в методі Віттена-Бела.

Таблиця 2.

Експеримент для методу Катца.

	Ентропія	Перплексія
Експеримент 1	6,269	77,1
Експеримент 2	6,349	81,5
Експеримент 3	6,3425	81,15
Експеримент 4	6,3446	81,26

Також, добре видно, що у порівнянні з результатами отриманими для інших мов на досить різноманітних корпусах [1,3,5], корпус стенограм Верховної Ради України є досить хорошим, з порівняно невеликим значенням ентропії.

Хоча основні очікувані результати виконуються подібно до першого експерименту, проте виграш очевидно є малим. Різниця між перплексіями у випадках додаванням невеликої кількості n -грам (експерименти 3 та 4) напрочуд мала порівняно з експериментом 2 у цій серії експериментів.

Це пов'язано з трьома факторами. По-перше, в методі згладжування Віттена-Бела перерозподіляється маса всіх n -грам, не залежно від їх частоти, а у методі згладжування Катца – лише маса n -грам, що мають частоту рівну або меншу за рівень відтинання. По-друге, параметр, що визначає рівень з якого n -грами є достовірними звичайно вибирають рівним 5, а в цьому експерименті він був рівний лише 3. Це пов'язано з нестійкістю формули Гуда-Тьюринга на даних зібраних з корпуса. Таким чином, в даному експерименті, n -грами з частотою 1 переважно виконували функцію джерела додаткової імовірнісної маси, а не значущих елементів.

По-третє, в методі Віттена-Бела величина, на яку відбувається зменшення значення частоти тієї чи іншої n -грами залежить від даних про всі інші n -грами, що починаються з $n-1$ однакових з нею слів. Таким чином, проста редукція не дає можливості коректно оцінити значення коефіцієнтів зменшення.

Висновки

Проаналізовано можливість застосування інформації про морфолексичні та граматичні характеристики слів для оптимізації моделей мови з метою покращення розпізнавання. Чисельні експерименти показали, що проста редукція, хоч і покращує потенційний результат розпізнавання, але не так сильно, як би того хотілося.

Подальші дослідження повинні бути спрямовані на пошук способу використання інформації про n -грами на базі канонічних форм та граматичних класів з метою водночас як оцінки числа різних видів очікуваних n -грам з малими частотами, так і використання частини маси n -грам з високими частотами для перерозподілу цієї маси серед очікуваних n -грам з малими частотами.

Список використаних джерел

1. S. F. Chen An empirical study of smoothing techniques for language modeling. /S. F. Chen, J. T. Goodman // Computer Speech and Language 1999, 13, –PP. 448-453.
2. D. Jurafsky Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition //Daniel Jurafsky, James H. Martin // Prentice Hall PTR Upper Saddle River, NJ, – 2000, ISBN:0130950696, – 934P.
3. Babyn D.N. On the prospects of creating a system of automatic continuous speech recognition for spoken Russian. [in Russian] / D.N. Babyn, I.L. Mazurenko, A.B. Holodenko // Intelligent systems, – vol.8 , issue 1-4, –2004. –PP. 45-70.
4. Kipyatkova I.S. The use of syntactic analysis to create n -gram language model for the Russian speech recognition systems [in Russian]/ I.S. Kipyatkova // Proceedings of the 5-th Interdisciplinary Seminar Analysis of conversational Russian language AR3 – 2011, 25 – 26 august 2011, –StPb, – PP. 13-18.
5. S. Ostrogonac Language model reduction for practical implementation in LVCSR systems / S. Ostrogonac, B. Popović, M. Sečujski, et al. // Infotech-Jahorina Vol. 12, March 2013. – PP. 391-394.

Надійшла до редколегії 27.09.2013

УДК 519.68

Терещенко В.М.¹, д.ф.-м.н., доц.

Алгоритми морфінгу полігонів у 2D

В роботі пропонується оптимальний алгоритм природнього морфінгу та модифікація інтуїтивного методу морфінгу полігонів на площині.

Ключові слова: морфінг, трансформація, полігон, інтерполяція.

¹ Київський національний університет імені Тараса Шевченка, 03680, м. Київ, пр-т. Глушкова 4д, e-mail: vtereshch@gmail.com

V.M. Tereshchenko¹, Dr. Sci.(Phis.-Math.), Assoc. Prof.

Algorithms morphing polygons in 2D

The paper proposes an optimal algorithm for natural morphing and modification of an intuitive method of morphing for polygons in the plane.

Key Words: morphing, transformation, polygon, interpolation.

¹ Taras Shevchenko National University of Kyiv, 03680, Kyiv, Glushkova av., 4d, e-mail: vtereshch@gmail.com

Статтю представив д.ф.-м.н., проф. Анісімов А.В.

Вступ

Постановка проблеми. Одним з важливих класів задач у комп'ютерній графіці є трансформація об'єктів (морфінг або перетворення). Алгоритми морфінгу мають чисельні застосування в інтерполяції фігур, анімації, відео компресії, а також просто як візуальний ефект. Існує багато досліджень в цій області, і всі методи діляться на дві основні групи: морфінг многокутників та морфінг зображень.

Однією з проблем морфінгу є реалістичність та природність перетворення об'єктів. Наступною проблемою є автоматизація цього процесу. Деякі алгоритми потребують завдання користувачем деяких ключових точок, відносно яких алгоритм буде виконувати трансформацію. Такі дії є небажаними, і звичайно, хороший алгоритм повинен здійснювати морфінг без будь-якого втручання користувача. При перетворенні простих многокутників (многогранників) постає ще одна проблема – це уникання самоперетину фігур під час морфінгу. Проблема самоперетину можна узагальнити до проблеми збереження деяких властивостей многокутника (многогранника) під час морфінгу. Щоб знайти ефективне розв'язання зазначених проблем, необхідно провести аналіз існуючих підходів.

Аналіз останніх досліджень. Аналізуючи існуючі підходи можна виділити два напрямки досліджень морфінгу: морфінг многокутників (многогранників) із заданими наперед

властивостями (паралельність ребер, рівна кількість кутів чи ребер) та морфінг многокутників (многогранників) у загальному випадку.

У роботах [1,2] розглядається задача трансформації простих многокутників P і Q , відповідні ребра яких взаємно паралельні і обходяться за годинниковою стрілкою, а також мають одну і ту ж орієнтацію. В [1] показано, що за $O(n^{4/3+\epsilon})$ кроків перетворення можна конвертувати один многокутник в інший. Пізніше ці результати були покращені Guibas та Hershberger [2], які досягли показника в $O(n \log n)$ кроків. Задача побудови простих многокутників з заданою послідовністю кутів має глибоку історію в обчислювальній геометрії. Culberson і Rawlins [3] перші підняли це питання і запропонували алгоритм, який будує такі многокутники за час $O(nD)$, де передбачається, що всі кути є раціональними множниками числа P_i , знаменники яких обмежені D . Vijayan і Wigderson покращили цей результат шляхом побудови таких полігонів за лінійний час без обмежень на кути [4]. Результат цього дослідження показує, що усі прості многокутники, створені з однієї послідовності кутів є гомотопними в цьому класі полігонів. Цей результат є цікавим, але немає формального доведення. Зауважимо, що для прямолінійних многокутників задача простіша і розв'язок уже відомий [5].

Один із підходів розв'язання задачі морфінгу базується на зведенні обох многокутників (многогранників) до спрощеної форми. Так, наприклад, в [1] представлено алгоритм, який працює шляхом перетворення кожного многокутника P і Q в спрощену форму. Ми спочатку перетворюємо P в його спрощену форму, потім спрощену форму P в спрощену форму Q , а потім перетворюємо в Q . Спрощена форма може бути отримана з P за $O(n)$ елементарних паралельних зміщень. Ієрархічна природа спрощеної форми природно може бути представлена у вигляді дерева. Проте, варто зазначити, що перетворення у спрощену форму є непривабливим з практичної точки зору, оскільки це може суттєво змінити масштаб різних особливостей оригінальних многокутників.

Цифрова деформація і морфінг мають глибокі корені в комп'ютерній графіці і повний виклад цих проблем виходить за межі цієї роботи. В [5] детально описується основні роботи в області 2D-морфінгу. Обчислення морфінгу многокутників може бути розділено на дві частини: визначення відповідностей між відповідними вершинами двох многокутників та визначення переходів. Слід вказати, що початковий і кінцевий многокутники не обов'язково мають одну і ту ж саму кількість вершин.

Задача пошуку відповідностей описана в [6], використовуючи фізичну модель. Ребра многокутників моделюються як еластичні зв'язки і трансформація фігур використовує обчислення фізичної взаємодії для мінімізації енергії системи. Цей алгоритм є ефективним для подібних вхідних многокутників і також може працювати у випадках, коли вхідні многокутники є повернутими чи перенесеними. Алгоритм не може бути застосований на многокутниках, які мають самоперетини. Обчислення траєкторій описано у роботі [7], де довжини ребер чи внутрішні кути інтерпольовані замість позицій вершин. Многокутники перетворюються в так звані ребро-кут представлення [5], що є інваріант до жорстких трансформацій. Ця схема інтерполяції дозволяє уникати руйнування ребер і немонотонної зміни кутів. Поняття інтерполяції параметрів також використовувалось пізніше для морфінгу плоских триангуляцій в [8, 9]. Ідеальна інтерполяція уникає локальних самоперетинів. Проте це не вирішує проблему глобальних самоперетинів, які можуть виникнути для дуже різних і складних фігур.

Shapira і Rappoport [10] розробили метод, який спочатку розбиває початковий і кінцевий многокутники на зіркові фігури, а тоді будує скелет – планарний граф, який з'єднує вершини зіркових полігонів з сусідніми полігонами. Скелети інтерпольовані, а з них відновлюються проміжні фігури. Різниця між цим підходом і методами описаними в [7] в тому, що цей підхід розглядає внутрішню область многокутника а не лише його границю. Проблема з цим підходом полягає в тому, що він спирається на ізоморфну декомпозицію в зіркові фігури, які можуть мати складнощі в обчисленні, особливо у випадку розрізаних фігур.

Alexa та інші [11,12] вводили інтерполяцію фігур «настільки жорстку, наскільки можливо». Вони обчислюють сумісні триангуляції вхідних многокутників. Сумісна триангуляція – розтин вхідного і вихідного многокутника так, щоб триангуляції були ізоморфними, тобто, існує одна взаємооднозначна відповідність між трикутниками у вхідному і вихідному многокутниках. Тоді може бути обчислена афінна трансформація, яка трансформує вхідні трикутники у вихідні. Інтерполяція в афінну трансформацію визначає морфінг. Аналогічні підходи також описані Surazhsky і Gotsman в [13]. Основна проблема цих методів полягає в обчисленні ізоморфних триангуляцій у вхідних фігурах, так як їхня якість впливає на якість морфінгу.

Gomez та інші ввели 2D-злиття [14], яке являється двомірним додатком до алгоритму, який був розроблений для 3D сіток. Спочатку вхідний многокутник відображається в одиничному колі, потім два відображення зливаються. Використовуючи обернене відображення, вершини початкового многокутника відображаються у другому многокутнику і навпаки. Це приводить до многокутників з однаковою кількістю вершин. Лінійна інтерполяція використовується, щоб отримати результуючий морфінг. Ця методика підходить лише до опуклого та зіркового многокутників.

Carmel і Cohen-Or [15] показали алгоритм, який поєднує 2D - злиття і розгортання многокутника. Користувач спочатку задає точки прив'язки, які визначають відповідність між многокутниками, а тоді обчислюється функція деформації так, щоб деформувати початковий полігон в кінцевий. Як тільки початковий многокутник стає деформованим, використовується метод розгортання

многокутника, щоб розгорнути початковий і кінцевий многокутник до опуклої фігури.

Johnstone і Wu [16] описали альтернативний метод злиття многокутників для морфінгу. 2-в-1 морфінг – фундаментальний випадок в морфінгу між різною кількістю многокутників. Основна ідея в тому, щоб злити два многокутники в один і потім використовуючи взаємоднозначну методику морфінгу многокутників, трансформувати між злитим многокутником і кінцевим полігоном. Під час злиття двох многокутників, ми трансформуємо кожен з них, доки вони не зустрінуться в точці. Тоді використовується методика розгортання кривої для морфінгу двох полігонів, які з'єднані в точці деякої великої фігури, яка потім трансформується до кінцевої фігури.

Метою роботи є розробка алгоритмів реалістичного морфінгу 2D об'єктів, з можливою адаптацією їх на тривимірний простір. Проект складається з розробки алгоритмів для створення плавного переходу між об'єктами трансформації (рівномірна інтерполяція), а також інтерфейс.

Для реалізації реалістичного морфінгу в роботі пропонується модифікація відомого інтуїтивного алгоритму [17] та новий підхід, який завдяки представленню об'єктів трансформації у вигляді три-гілкового бінарного дерева дозволяє здійснювати лінійний морфінг. Ми пропонуємо таку стратегію розв'язання задачі морфінгу, яка б дозволяла автоматично здійснювати реалістичну трансформацію об'єктів, зберігаючи пропорції об'ємів та кривизну поверхні. Така стратегія повинна поєднувати в собі покращений спосіб співставлення з можливістю виявлення упорядкованих точок для автоматизації пошуку відповіностей та більш реалістичну техніку інтерполяції, ніж класична лінійна.

1 Інтуїтивний алгоритм морфінгу многокутників

В роботі [17] представлено алгоритм інтуїтивного морфінгу многокутників. Алгоритм представлений в трьох варіаціях. Один з них підходить для опуклих многокутників, а інші два для більш складних полігонів, таких як зігнуті чи спірально-закручені многокутні форми.

Нехай маємо многокутники A і B . Розглянемо, наприклад перетворення $A \rightarrow B$. Зводимо центроїди цих многокутників в одну точку O . Одержимо, у загальному випадку, два многокутники які перетинаються, рис.1. Область перетину многокутників $C=A \cap B$ задає так зване

«ядро» морфінгу. Морфінг одночасно змінює два многокутники. Вхідний многокутник абсорбований ядром, тоді як вихідний многокутник росте з нього. При цьому алгоритм дає задовільні результати для многокутників, які не являються зірковими. Алгоритм знаходить відповідності між двома вхідними многокутниками як шляхи вершин. Кінцеве перетворення є лінійна інтерполяція між проміжними позиціями в шляхах вершин. Без втрати загальності, нехай перетин складається з одного простого многокутника.

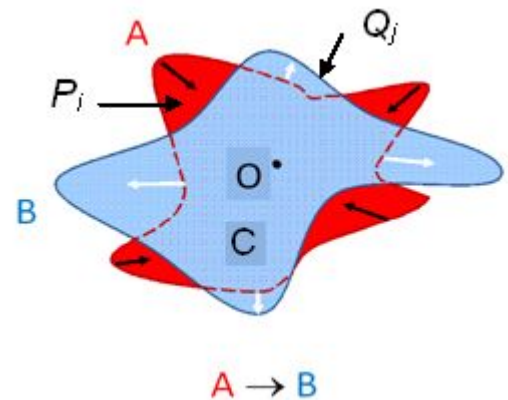


Рис. 1. Модель 2D - морфінгу. A і B – вхідні многокутники. C – ядро. P_i та Q_j – многокутники, які виростають з ядра чи абсорбуються в нього.

Область многокутника A , відрізану ядром, позначимо $P=A-B$ та аналогічну область многокутника B позначимо як $Q=B-A$. Зазначимо, що P і Q можуть бути як одиничними полігонами так і множинами простих многокутників: $P=UP_i$ та $Q=UQ_j$. Припустимо, що

$P \neq \emptyset$ чи $Q \neq \emptyset$. Під час процесу морфінгу частини

Q_j виростають з ядра, в той час як частини P_i абсорбуються ядром. Всі частини многокутників ростуть і абсорбуються одночасно, що призводить до ефекту морфінгу двох простих многокутників. Алгоритмічно, процес абсорбції – зворотній до процесу росту, тому ми поки що зосередимось тільки на описі процесу абсорбції однієї частини P_i . Розглянемо декілька відомих методів росту.

Метод росту по периметру. Початкова ідея методу полягає в тому, що вершини $v_i \in C_{out}$ рухаються вздовж вихідного ланцюга C_{out} многокутника P_i , зважаючи на те, що шлях вершини містить тільки позиції вершин δP_i . Коли

будуть відомі відповідності між вершинами вхідного ланцюга C_{in} та вихідного ланцюга C_{out} , ми повинні визначити для кожної вершини v_j топологічну відстань d_j в шляху до кінця. Шлях вершини v_i з топологічною відстанню d_i містить вершини з топологічними відстанями $(d_{i-1}, d_{i-2}, \dots, d_0, d_{-1}, \dots, d_j)$ (рис.2, [17]). Ми задаємо дві умови відносно останньої вершини в шляху вершини v_j . По-перше, вона повинна належати ланцюгу C_{in} . По-друге, повинен існувати, як мінімум, один шлях вершини, який закінчується в кожній вершині, яка належить ланцюгу C_{in} . На підставі цих спостережень, можна сказати, що шлях вершини з d_{max} завжди закінчується у вершині з d_{min} . Якщо існує тільки одна вершина з d_{max} , ми подвоюємо її, щоб одержати розв'язок для випадку двох вершин з d_{min} . Шлях вершини з топологічною відстанню d_{max-1} повинен закінчитися у вершині з d_{min+1} . В загальному випадку, шлях вершини d_{max-i} повинен закінчитися у вершині з d_{min+i} , як показано на рис.2 [17].

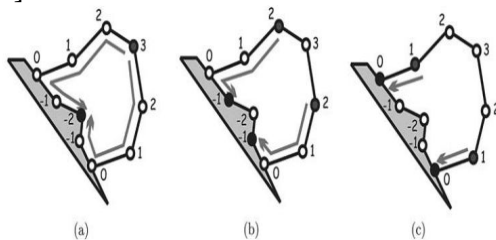


Рис.2. Метод росту серединної точки

Метод росту серединної точки. На відміну від попереднього методу в методі росту серединної точки, ріст серединної точки намагається зробити верхню вершину за допомогою росту меншої частини, ніж у рісті по периметру. Цей метод використовує середини сегментів ліній, визначених вершинами ланцюгів C_{in} чи C_{out} з одними і тими ж топологічними відстанями.

Метод росту по проекції. Попередні методи росту можуть генерувати небажані деформації на кінцях вирощеної частини – формально вони «вирізають» кінцеву точку, подальші згини кінцевої точки незначні, але ні один з методів не показує частини кінцевої фігури в цій точці. Тому ці два методи дають задовільні результати у випадку вузьких чи кривих частин морфінгу. До частин морфінгу, які не зігнуті, був створений новий метод, який заснований на проекції і називається методом росту проекції [17]. Цей метод рівновіддалено відображає вершини на лінійний сегмент, який з'єднаний з двома вершинами на перетині. Головна відмінність

методу росту проекції від двох інших полягає в тому, що усі вершини з C_{out} мають одну і ту ж кількість елементів в їхніх шляхах.

2 Запропоновані модифікації морфінгу полігонів

В роботі пропонується оптимальний алгоритм природнього плоского морфінгу полігонів із збереженням об'ємних пропорцій та модифікація методу росту по периметру. Спочатку розглянемо модифікацію. Для підвищення ефективності і покращення зручності застосування в розглядуваній роботі пропонується модифікація методу росту по периметру.

2.1 Модифікований метод росту по периметру

Ідея методу полягає в тому, що при збереженні головної суті методу росту по периметру, орієнтація вершини з топологічною відстанню 0 через яку проходить шлях, обчислюється в залежності від довжини шляху. Ця модифікація буде працювати краще у випадках, коли вершини лівих і правих частин зовнішнього ланцюга розміщені на різних відстанях. В такому випадку рідше можливий випадок самоперетину під час морфінгу. Останнім кроком алгоритму є злиття. Многокутники P_i і Q_i , які були відрізані ядром C , повинні бути злиті в один многокутник R . Окрім того, кожна вершина в R має свої шляхи вершин, які визначені одним з вищеописаних методів. Напрямок кожної вершини залежить від напрямку морфінгу. Процес злиття був описаний алгоритмом Weiler–Atherton для перетину многокутників [4]. Він обробляє кожний многокутник і робить копії вершин з додатною топологічною відстанню до нового списку вершин, зберігаючи при цьому порядок вершин.

2.2. Оптимальний алгоритм природнього морфінгу полігонів

Природній морфінг геометричних об'єктів полягає в реалістичному рівномірному перетворенні одного полігону в інший без перекоєнень та самоперетинів. При цьому передбачається збереження пропорційності форм (об'ємів). Для забезпечення природньої трансформації полігонів в роботі [17] запропоновано так званий «інтуїтивний алгоритм росту полігонів». Цей алгоритм дає бажаний результат, проте, щодо оцінки складності, він не є найкращий. В представлений роботі

пропонується алгоритм, який забезпечує якість перетворення об'єктів інтуїтивного алгоритму, але з суттєвим підвищенням ефективності реалізації. Опишемо основні кроки алгоритму.

Алгоритм

Нехай задано два полігони (многокутники) A і B з m та k вершинами, відповідно рис. 3,4.

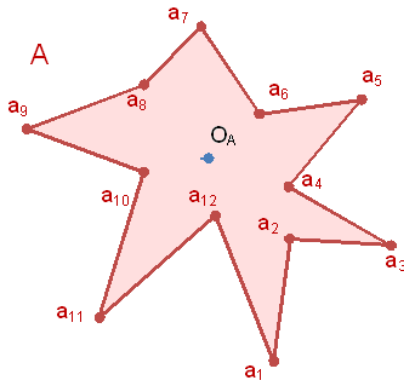


Рис.3. Полігон А.

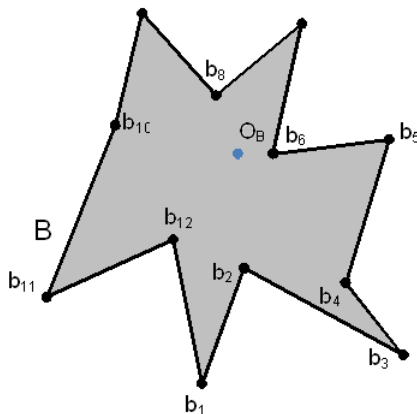


Рис.4. Полігон В.

Попередня обробка

1. Спочатку визначимо центроїди многокутників A і B , $O_A(x_A, y_A)$ та $O_B(x_B, y_B)$, відповідно: $x_A = \sum_{i=1}^m x_i / m$, $y_A = \sum_{i=1}^m y_i / m$, $x_B = \sum_{j=1}^k x_j / k$, $y_B = \sum_{j=1}^k y_j / k$.
2. Стягуємо центроїди обох многокутників в одну точку O . Одержимо Два многокутники, які, в загальному випадку, перетинаються, рис. 5.

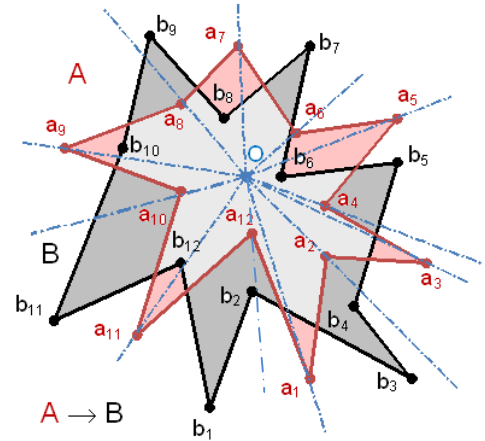


Рис.5. Перетворення А в В.

3. Відсортуємо вершини обох многокутників відносно одержаного центроїда O за полярним кутом в напрямку проти годинникової стрілки. Це робиться просто так, як обидва многокутники представлені у вигляді циклічно впорядкованих списків вершин.
4. Одержимо два впорядкованих списки: $A = \{a_1, a_2, \dots, a_m\}$, $B = \{b_1, b_2, \dots, b_k\}$

Перетворення

Нехай, не обмежуючи загальність, полігон A перетворюється в B .

5. Проводимо промені з точки O до вершин многокутника A . Одержимо впорядковану множину секторів, які відповідають списку A .
6. Початкові елементи списків A і B обираємо так, щоб вони починались із сусідніх за кутом вершин (наприклад найменші у-координаті вершини). У нашому прикладі на рис. 5 це вершини a_1 , b_3 .
7. Рухаємось по списку A , розглядаючи сектори визначені двома сусідніми вершинами списку (a_i, a_{i+1}) . Для кожної такої пари рухаючись по списку обираємо вершини і ребра многокутника B , які потрапляють в сектор, визначений цією парою.
8. Таким чином рухаючись паралельно по спискам A і B для кожного ребра многокутника A визначаємо елементи многокутника B в які перетворюємо ребро з A .

9. Обійшовши паралельно списки А і В в результаті інтерполяції відповідних елементів многокутників одержимо перетворення А в В за лінійний час.

3 Програмна реалізація

На практиці реалізовано модифікаційний інтуїтивний алгоритм росту по проекції та оптимальний алгоритм природнього морфінгу простих полігонів. Для ілюстрації результатів у роботі представлено реалізацію лише першого алгоритму, рис. 6.

Реалізація виконана у вигляді Windows-програми у середовищі Borland Delphi 7 з використанням програмного інтерфейсу OpenGL. У вікні відобразиться анімація морфінгу з одного полігона в інший.

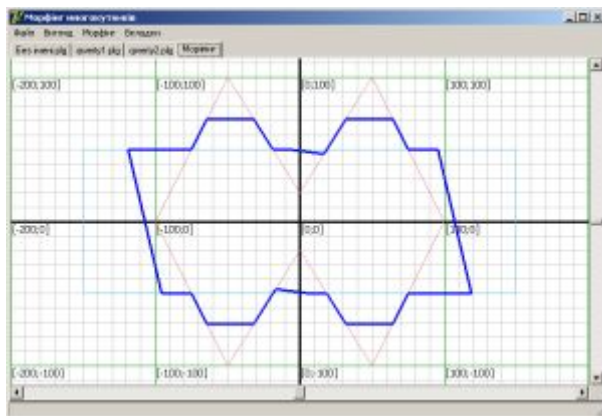


Рис.6. Приклад модифікованого інтуїтивного морфінгу полігонів.

Список використаних джерел

1. Guibas L.J., Hershberger J. Morphing simple polygons // Proc. 10th Annu. Sympos. Comput. Geom. – 1994. – P. 267–276.
2. Hershberger J., Suri S. Morphing Binary Trees//Proc. 6th ACM-SIAM Sympos Discrete Algorithms. – 1995. – P. 396–404.
3. Culbertson J., Rawlins G. Turtlegons: generating simple polygons from sequences of angles // Proc. of the first ann. Symp. on Comput. Geom. – 1985. – P. 305–310.
4. G. Vijayan. Geometry of planar graphs with angles // Proc. of the second ann. Symp. on Comput. Geom. – 1986. – P. 116–124.
5. Thomassen C. Deformations of planar graphs. // J. of Combinatorial Theory. – 1983. – S. B. – V. 34 (3). – P. 244–257.
6. Sederberg, T.W., Greenwood, E. A physically based approach to 2-d shape blending //J. Comput. Graphics. –1992. – 26. – P. 25–34.
7. Sederberg, T.W., Gao, P., Mu, G., Wang. 2-d shape blending: An intrinsic solution to the vertex path problem //J. Comput. Graphics. – 1993. – 27. – P. 15–18.
8. D. Sleator, R. Tarjan, and W. Thurston. Rotation distance, triangulations, and hyperbolic geometry// J. of the American Mathem. Soc. – 1988. – 1(3). –P. 647–681.
9. Graf T., Veezhinathan K. Reducing simple polygons to triangles // Proc. 25th Int. Colloq. on Automata Lang. and Program., Lecture Notes in Comp. Scien. – 1998. – V. 1. – P. 130–139.
10. Shapira M., Rappoport A. Shape blending using the starskeleton representation // IEEE Comput. Graph. Appl. –1995. –15. – P. 44–50.
11. Alexa M., Cohen-Or D., Levin D. Asrigid-as-possible shape interpolation // Computer Graphics Proceedings, ACM SIGGRAPH. – 2000. – P. 157-164.
12. Alexa M. Recent advances in mesh morphsubmitted ing // Computer Graphics Forum. – 2002. – 21 (2). –P. 173-196.
13. Surazhsky V., Gotsman C. Intrinsic morphing of compatible triangulations // Int. J. Shape Model. – 2003. –9. –P. 191–201.
14. Gomes J. Warping and Morphing of Graphical Objects / Morgan Kaufmann, San Mateo. –1999.
15. Carmel E., Cohen-Or D. Warp-guided object-space morphing //Vis. Comput. – 1997. – 13. – P. 465–478.
16. Johnstone, J.K., Wu, X. Morphing two polygons into one // The 40th Ann. Southeast ACM Conference. – 2002. – P. 56-61.
17. Málková M., Parus J., Kolingerová I., Beneš B. An Intuitive Polygon Morphing // The Visual Computer, Springer Verlag. – 2010. – 26(3). P. 205-215.

Надійшла до редакції 12.09.13

такою, що задача (4)-(6) має n обмежень та m змінних, матриця A^T обмежень витягнута вертикально та подається у вигляді

$$\max Bu, \quad (4)$$

за умов:

$$A^T u \leq C^T, \quad (5)$$

$$u_i \geq 0, \quad (i = \overline{1, m}) \quad (6)$$

Положення методу псевдобазисних матриць

Підматрицю A_0 матриці A^T , складену із m лінійно незалежних нормалей $J_b = (i_1, i_2, \dots, i_m)$ обмежень (5), будемо називати *псевдо базисною*, а розв'язок $u_0 = (u_{01}, u_{02}, \dots, u_{0m})$ відповідної їм системи рівнянь $A_0 u_0 = C^0$, де $C^0 = (c_{i_1}, c_{i_2}, \dots, c_{i_m})$ - підвектор C псевдо базисним при виконанні умови $\alpha_{0i} \geq 0, i = \overline{1, n}$.

Нехай: $\beta_{ij}, i, j \in I = \{1, 2, \dots, m\}$ - елементи A_0 ; e_{ri} та $(A_0^{-1})_i$ елементи та i -й стовпець A_0^{-1} , оберненої до A_0 ; $\alpha_r = (\alpha_{r1}, \alpha_{r2}, \dots, \alpha_{rm})$ - вектор розвинення нормалі $a_r u_1 \leq c_r$ за рядками A_0 , $\alpha_0 = (\alpha_{01}, \alpha_{02}, \dots, \alpha_{0m})$ вектор розвинення нормалі цільової функції (4) за рядками A_0 ;

$\Delta_r = a_r u_0 - c_r$ - нев'язка r -го обмеження (5), а $\Delta_0 = Bu_0$ - значення цільової функції в вершині u_0 , які утворюють вектор $\Delta = (\Delta_0, \Delta_1, \dots, \Delta_n)$; J_0, J_n - множини індексів, відповідно базисних і небазисних обмежень (5).

Всі означені елементи при переході до суміжної $\bar{A}_0$, яка утворюється із A_0 заміною її рядка a_k на a_l , що не входить в A_0 будемо позначати рискою зверху, тобто $\bar{\beta}_{ij}, \bar{\alpha}_r, \bar{L}_i, \bar{\Delta}_k, \bar{e}_{ri}, (\bar{A}_0^{-1})_i, \bar{\alpha}_0$.

Нехай $a_{i_1}, a_{i_2}, \dots, a_{i_m}$ - нормалі, $a_j u \leq c_j, j \in J_0$ де $J_0 = \{i_1, i_2, \dots, i_m\}$ - індекси обмежень, нормалі яких утворюють A_0 , a_l - вектор-нормалі $a_l u \leq c_l, \alpha_l = (\alpha_{l1}, \alpha_{l2}, \dots, \alpha_{lm})$ - вектор розвинення a_l за рядками A_0 .

Теорема 1. [4] Між коефіцієнтами розвинення нормалей обмежень (5) та цільової

функції (4) за рядками базисної матриці, елементами обернених матриць, базисними розв'язками, нев'язками обмежень (5) та значеннями цільової функції в двох суміжних базисних розв'язках мають місце такі співвідношення

$$\bar{\alpha}_{rk} = \frac{\alpha_{rk}}{\alpha_{lk}}, \quad \bar{\alpha}_{ri} = \alpha_{ri} - \frac{\alpha_{rk}}{\alpha_{lk}} \alpha_{li},$$

$$r = \overline{0, n}; \quad i = \overline{1, m}; \quad i \neq k; \quad (7)$$

$$\bar{e}_{rk} = \frac{e_{rk}}{\alpha_{lk}}, \quad \bar{e}_{ri} = e_{ri} - \frac{e_{rk}}{\alpha_{lk}} \alpha_{li},$$

$$r = \overline{1, m}; \quad i = \overline{1, m}; \quad i \neq k; \quad (8)$$

$$\bar{u}_{0j} = u_{0j} - \frac{e_{jk}}{\alpha_{lk}} \Delta_l, \quad j = \overline{1, m}, \quad (9)$$

$$\bar{\Delta}_k = -\frac{\Delta_l}{\alpha_{lk}}, \quad \bar{\Delta}_r = \Delta_r - \frac{\alpha_{rk}}{\alpha_{lk}} \Delta_l,$$

$$r = \overline{1, n}; \quad r \neq k; \quad (10)$$

$$\bar{B}u_0 = Bu_0 - \frac{\alpha_{0k}}{\alpha_{lk}} \Delta_l, \quad (11)$$

причому умовою: опорності псевдобазисності матриці при вводити вектора нормалі a_l обмеження $\alpha_l u \leq c_l$ на k -ту позицію базисної матриці A_0 є $\alpha_{lk} \neq 0$; псевдобазисності опорних розв'язків, тобто $\alpha_{0i} \geq 0, i = \overline{1, n}$ є $\alpha_{lk} > 0$: спадання цільової функції у новій псевдобазисній матриці при задачі на максимум є $\Delta_l > 0$, зростання цільової функції $\Delta_l < 0$, сталості $\Delta_l = 0$.

Встановлено [5], що якщо існує базисна матриця A_0 така, що $\alpha_{0k} \geq 0, k = \overline{1, m}$, то базисна матриця та відповідний їй розв'язок u_0 оптимальні, причому при $\alpha_{0k} > 0, k = \overline{1, m}$, розв'язок єдиний, при $\exists i_0, \alpha_{0i_0} = 0$, розв'язок неєдиний.

Для матричної гри у змішаних стратегіях ці положення для елементів методу набудуть наступного вигляду [5].

Твердження 1. Компоненти вектора розкладу цільової функції задачі (4)-(6) в ході ітерацій методу базисних матриць обчислюються за формулами

$$\alpha_{0i} = \sum_{j=1}^m e_{ji}, \quad i = \overline{1, m},$$

причому для оптимальності розв'язку необхідно та достатньо, щоб суми елементів стовпців оберненої матриці були невід'ємними

$$\alpha_{0i} = \sum_{j=1}^m e_{ji} \geq 0, i = \overline{1, m},$$

для єдиності оптимального розв'язку (матричної гри) необхідно та достатньо щоб суми елементів стовпців оберненої матриці були строго додатними

$$\alpha_{0i} = \sum_{j=1}^m e_{ji} > 0, i = \overline{1, m},$$

тобто $S_1^c > 0, S_2^c > 0, \dots, S_m^c > 0$),)

при $\exists k_0, \alpha_{k_0} = 0, (S_{k_0}^c = 0)$, розв'язок неєдиний.

Твердження 2. Компоненти вектора розв'язків задачі (4)-(6) в ході ітерацій методу базисних матриць обчислюються за формулами

$$u_{0i} = \sum_{j=1}^m e_{ij}, i = \overline{1, m}.$$

Наслідок 1. Якщо $S_1^r, S_2^r, \dots, S_m^r$ - суми елементів стовпців оберненої матриці $A_b^{-1(r)}$ на r -ій ітерації набувають невід'ємних значень

$$(S_1^r \geq 0, S_2^r \geq 0, \dots, S_m^r \geq 0),$$

то дана обернена базисна матриця є оптимальною, а відповідний базисний розв'язок є оптимальним, причому його компоненти знаходяться за співвідношенням

$$u_{0i} = \sum_{j=1}^m e_{ij}, i = \overline{1, m}$$

(суми елементів рядків оберненої матриці).

Наслідок 3. Сума елементів рядків оберненої матриці співпадають із значенням відповідної компоненти вектору проміжного розв'язку на ітераціях методу базисних матриць.

Доведення. Справедливість тверджень є наслідком властивостей розкладу елементів методу базисних матриць за рядками базисної матриці.

Неважко переконатись, що при $C = (\underbrace{1, 1, \dots, 1}_n); B = (\underbrace{1, 1, \dots, 1}_m)^T$ та відомими

$$A_b, A_b^{-1}$$

$$\alpha_0 = B \times A_b^{-1}, u_0 = A_b^{-1} \times C_b$$

для формул зв'язку елементів методу базисних матриць (7)-(9) будуть справедливі наступні співвідношення

$$\alpha_{0i} = \sum_{j=1}^m e_{ji}, i = \overline{1, m}, u_{0i} = \sum_{j=1}^m e_{ij}, i = \overline{1, m}.$$

При розв'язанні двоїстої пари задач лінійного програмування (матричної гри у

змішаних стратегіях) на основі методу базисних матриць можна встановити ряд властивостей

Схема методу псевдобазисних матриць для матричної гри у змішаних стратегіях

1. Якщо існує псевдобазисна матриця A_b , що розвинення за її рядками вектору цільової функції визначається як

$$\alpha_{0k} = S_k^c = \sum_{j=1}^m e_{jk} \geq 0, k = \overline{1, m},$$

$$\Delta_r \leq 0, r \in J,$$

то дана псевдобазисна матриця є оптимальною, а відповідний їй псевдобазисний розв'язок оптимальним, причому при

$$\alpha_{0k} = S_k^c = \sum_{j=1}^m e_{jk} > 0, k = \overline{1, m} \text{ розв'язок єдиний,}$$

а при $\exists k$ такого, що $\alpha_{0k} = S_k^c = \sum_{j=1}^m e_{jk} = 0$, то розв'язок неєдиний.

2. Якщо існує псевдобазисна матриця A_b та індекс l такий, що $\Delta_l > 0$, для якого $a_{li} > 0$, а

$$\text{також індекс } k \text{ такий, що } -\frac{\alpha_{0k}}{\alpha_{lk}} = \min_{\substack{i, \\ a_{li} > 0}} \frac{\alpha_{0i}}{\alpha_{li}}, \text{ то при}$$

заміні рядка k на l нова матриця $\bar{A}_b$ буде псевдобазисною. При цьому можна перейти до нового псевдобазисного розв'язку, в якому значення цільової функції зменшиться на

$$-\frac{\alpha_{0k}}{\alpha_{lk}} \Delta_l.$$

3. Якщо існує A_b , в якому $a_{ok} \geq 0, k = \overline{1, m}$, $a_r u_0^T \leq c_r$ де u_0 псевдовершина, що відповідає A_b , то обмеження $a_r u^T \leq c_r$ пасивне (несуттєва чиста стратегія) і може бути виключене із обчислень.

Висновок

На основі тверджень, що обґрунтовують МПБМ, можуть бути запропоновані різноманітні алгоритмічні схеми проведення аналізу та оптимізації лінійної системи. Слід зазначити, що структурні особливості задачі (4)-(6) такі як наявність $0 \notin E^m$ та відповідної псевдобазисної матриці сприятливі для застосування МПБМ до цієї задачі (початкова псевдобазисна матриця формується на основі нормалей обмежень, що

формуують півпростори додатнього ортанту.

Врахування структурних властивостей обмежень задачі (2)-(3) дозволяє: побудувати двосторонні оцінки значень змінних та цільової

функції (1), локалізувати область належності оптимальних значень, додатково ідентифікувати пасивні та неактивні обмеження (стовпці), застосувати другу теорему двоїстості (про умови доповнюючої нежорсткості) для виявлення суттєвих та несуттєвих стратегій.

Список використаних джерел

1. *Golshteyn E.G., Yudin D.B.* New directions in linear programming. – М. – Sovetskoe radio, – 1969, – 524p. (in Russian).
2. *Dantzig G.B., Thapa M.N.* Linear Programming 1: introduction, Springer, – 1997, – 435p.
3. *Golshteyn E.G., Yudin D.B.* Linear programming/.Theory and methods. –М.: Nauka, – 1963. – 776p. (in Russian).
4. *Kudin V. I., Lyashko S.I., Khritonenko N.V., Yatsenko Yu.P.* Analysis of the properties of a linear system using the method of artificial basis matrices // *Kibernetika i sistemny analiz.* — 2007. — N 4. —.P. 119–127 (in Ukrainian).
5. *Todorico B.D.* The analysis of matrix game with mix strategies how linear programming problems with application method of basic matrix // *Visnyk KNU.*-2013,-№3,-P. 3-7.

Надійшла до редколегії 05.04.13

УДК 004.021 + 004.023 + 515.125

Андрій Л. Фісуненко*, аспірант

Генерація простих багатокутників вирізанням та нарощуванням опуклих оболонки.

Запропоновано евристичний рекурсивний алгоритм генерації простих багатокутників для множини точок на площині і розглянуто його застосування та узагальнення для d-вимірного випадку.

Ключові слова: множина точок, простий багатокутник, генерація, опукла оболонка.

*E-mail: alf.via @ gmail.com

Статтю представив член-кор. НАН України, д.ф.-м.н., проф. Анісімов А. В.

Вступ

Постановка проблеми. Розглядається задача генерації простих багатокутників із заданими умовами. Наївне рішення простим перебором можливих перестановок та відкиданням варіантів з самоперетинами відрізків (*Permute & Reject*, складність $O(N!)$) для сучасних обчислювальних систем та таких, що очікуються в осяжному майбутньому, дозволяє вирішувати задачу за прийнятний час тільки для $N < 15$, що для більшості практичних застосувань є, очевидно, далеко недостатнім.

Аналіз останніх досліджень. Загальне питання, чи існує поліноміальний алгоритм для обчислення усіх можливих простих багатокутників, таких, що кожен з них включає всі N точок заданої множини S , наразі залишається відкритим (проблема 16 з переліку відкритих проблем обчислювальної геометрії) [1]. Зазначимо, що для будь-якої множини S із N точок на площині при $N > 2$, якщо усі вони не належать одній прямій, можливо побудувати простий багатокутник, вершинами якого будуть усі точки S . Аналогічне твердження доведене для E^3 [2]. Також відомо, що задача пошуку багатокутників мінімальної та максимальної площі на фіксованій множині точок S є NP-складною [3].

Традиційно актуальність побудови ефективних алгоритмів генерації простих багатокутників пов'язується як з теоретичним інтересом, так і з практичною необхідністю перевірки коректності та оцінок складності часу

Andriy L. Fisunenko, PhD student.

Generation of simple polygons by cutting out and building up convex hulls.

We propose a heuristic recursive algorithm for generating simple polygons on a given finite point set on the plane and discuss its applications and generalizations for the case of d-dimensional space.

Key Words: point set, simple polygon, generation, convex hull

виконання і використання пам'яті для багатьох алгоритмів обчислювальної геометрії.

Оскільки загальне питання про існування поліноміального алгоритму залишається відкритим, то для генерації простих багатокутників на практиці використовуються різного роду евристики [4, 5] та вивчаються алгоритми генерації окремих конфігурацій багатокутників із заданими властивостями: наприклад, спіральні [6, 7], квітко-подібні [8]. Для вирішення задачі мінімального багатокутника на площині використовуються наближені алгоритми [9].

Новизна та ідея. В статті запропоновано новий евристичний алгоритм побудови простих багатокутників, основна ідея побудови головної процедури якого полягає в наступному. На першому кроці знаходиться опукла оболонка $CH(S)$. Далі множина точок, що складається з фрагменту поточного контуру та потім, в залежності від кроку, зовнішніх чи внутрішніх «вільних» точок, охоплюється опуклою оболонкою з приєднанням її нового фрагменту до цього контуру. Тобто частина «вільних» точок, які опиняються на опуклій оболонці, приєднуються до багатокутника, що будується. Для решти «вільних» точок процес повторюється до тих пір, доки усі вони не опиняться у складі багатокутника.

Мета статті полягає у побудові наближеного методу пошуку простих багатокутників мінімальної (або максимальної) площі з використанням запропонованого алгоритму, а також огляду схеми доведення існування

простого многогранника для заданої множини точок для простору E^d .

Постановка задачі та алгоритм розв'язання

Постановка. Розглянемо множину точок $S = \{P_1, P_2, \dots, P_N\}$ ($N > 2$) на площині E^2 . Без втрати загальності будемо вважати, що усі точки знаходяться у загальній позиції, тобто жодні три точки, що належать S , не є колінеарними. Задача полягає в побудові простих многокутників на множині точок S за певним критерієм. Нас будуть цікавити такі прості многокутники $P = \{P_{\pi(1)}, P_{\pi(2)}, \dots, P_{\pi(N)}\}$, контури яких утворюються усіма точками множини S .

Алгоритм. Для спрощення, без втрати загальності, розглядаємо тільки варіант обходу многокутника проти годинникової стрілки, виключаючи варіанти, які можна отримати циклічними зсувами нумерації його точок або зміною напрямку обходу.

Основні кроки алгоритму, опускаючи несуттєві деталі:

1. Побудувати опуклу оболонку $CH(S)$ множини S (складність $O(N \log N)$) та прийняти її за поточний контур многокутника **polygon**, що будується, а також за поточний фрагмент контуру **contourFragmen** для наступного кроку, окремо зберегти множину внутрішніх «вільних» точок **freePoints** ($S \setminus CH(S)$) та контур **polygon**;
2. Виконати рекурсивну процедуру **doCutAndGrow** для **polygon**, **contourFragmen** та **freePoints**:
 - 2.1 Якщо всі точки множини S вже утворюють поточний контур многокутника **polygon**, вийти з рекурсії з кінцевим результатом;
 - 2.1.1 Цикл по всім можливим суб-фрагментам **subFragment**(i, j) поточного фрагменту контуру **contourFragmen**, де i та j – індекси точок, що утворюють опорні відрізки;
 - 2.1.2 Побудувати опуклу оболонку $CH(S_{ij})$ на точках **subFragment**(i, j) та **freePoints**, запам'ятовуючи нові «вільні» точки опуклої оболонки **localFreePoints**(i, j);
 - 2.1.3 Приєднати фрагмент $CH(S_{ij}) \setminus subFragment(i, j)$ до поточного контуру **polygon** многокутника, що

будується, приєднуючи точки між i та j , якщо вони не суміжні, до **localFreePoints**(i, j), та видаляючи їх з **polygon**.

- 2.1.4 Перевірити, а) що контур **polygon** не зустрічався раніше, б) перевірити **цільовий предикат відбору** для даного контуру та виконати рекурсивно процедуру **doCutAndGrow** для **polygon**, **subFragment**(i, j), **localFreePoints**(i, j), інакше перейти до наступного кроку циклу.

На Рис. 1 показано основні кроки алгоритму на прикладі множини з 18 точок. Пунктиром позначено опорні суб-фрагменти для послідовних рівнів рекурсії. В даному прикладі для спрощення використані суб-фрагменти без внутрішніх точок. На першому рівні рекурсії на суб-фрагменті P_1P_2 опуклої оболонки $CH(S)$ будується опукла оболонка $P_1, P_2, P_8, P_9, P_{10}, P_{11}, P_{12}$, а точки $P_{13}, P_{14}, P_{15}, P_{16}, P_{17}, P_{18}$ залишаються вільними. На другому рівні на суб-фрагменті P_2P_8 будується опукла оболонка $P_8, P_{13}, P_{14}, P_{15}, P_2$, вільні точки – P_{16}, P_{17}, P_{18} . Третій рівень рекурсії виконується на суб-фрагменті P_8P_{13} та вільних точках P_{16}, P_{17}, P_{18} , що дає оболонку $P_8, P_{13}, P_{16}, P_{17}$, та останню внутрішню точку P_{18} . Рекурсія завершується четвертим рівнем побудовою трикутника P_{13}, P_{16}, P_{18} на суб-фрагменті P_{13}, P_{16} .

В залежності від рівня рекурсії «вільні» точки по чергово опиняються зовні чи всередині многокутника, що будується. Тому умовно для вважаємо, що алгоритм «вирізає» чи «нарошує» контур опуклими оболонками, хоча рекурсивна процедура в обох випадках залишається універсальною.

Зауваження 1. Якщо алгоритм виконується тільки для суб-фрагментів, що утворюються сусідніми точками контуру, пропускається крок в 2.2.3, який для виключення зациклювання рекурсії перевіряє, чи контур не зустрічався раніше.

Зауваження 2. Цільовий предикат відбору – процедура, яка перевіряє певний критерій, який визначається задачею та за допомогою якого обирається один із можливих шляхів до наступного рівня рекурсії. Без цього критерію алгоритм у найгіршому випадку при $O(N)$ рівнів рекурсії має експоненціальну складність $O(N^{2N})$.

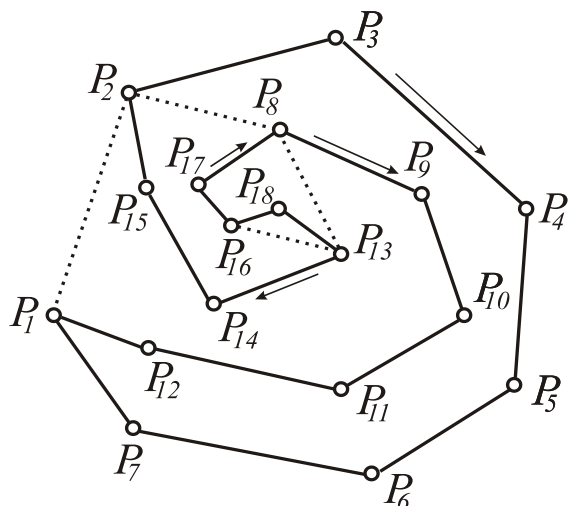


Рис. 1. Приклад роботи алгоритму вирізання та нарощування контуру опуклими оболонками.

Зауваження 3. Якщо складність цільового предикату $O(p(N))$, де p - деяка поліноміальна функція, складність перевірки на повторний контур $O(\log_2 N)$, використовується один із алгоритмів опуклої оболонки $O(N \log_2 N)$, глибина рекурсії $O(N)$, і перебираються усі пари суб-фрагментів i, j ($O(N^2)$) для пошуку варіанту, що відповідає цільовому предикату, то у найгіршому випадку алгоритм має складність $O(p(N)N^3(\log_2 N)^2)$.

Зауваження 4. Для окремих випадків, таких як спіральні многокутники (далі) або генерація випадкового многокутника цикл по суб-фрагментам $O(N^2)$ відсутній і складність нижча.

Спіральні многокутники

Запропонований алгоритм, як частинний випадок, покриває генерацію таких спіральних многокутників, які описані в роботах [6, 7]. У прикладі на Рис. 1, представлено генерацію саме такого многокутника (Рис.1). Для побудови спірального многокутника необхідно обирати тільки перші суб-фрагменти на кожному рівні рекурсії. Через те, що усі точки опуклих оболонок, що будуються алгоритмом, впорядковуються в однаковому напрямку (наприклад, проти годинникової стрілки) автоматично утворюється спіральна конфігурація.

Складність алгоритму в найгіршому випадку $O(N^2 \log_2 N)$.

Многокутник мінімальної площі

Для побудови наближеного алгоритму пошуку многокутника мінімальної площини використовуємо наступний підхід. На першому кроці будується опукла оболонка $CH(S)$. Далі обирається опорна точка P_0 , яка є внутрішньою точкою опуклої оболонки, але не співпадає з жодною точкою вхідної множини точок: $P_0 \notin S$. Будуємо трикутники, що утворюються кожною з граней опуклої оболонки і P_0 . Для кожного трикутника, що містить внутрішні точки виконуємо описаний вище алгоритм, використовуючи грань опуклої оболонки у якості єдиного початкового суб-фрагменту. На першому рівні рекурсії (крок «вирізання») перевіряємо умови мінімальності загальної площі і переходимо на вибір наступної опорної точки P_0 , якщо сукупна площа виявилася більшою ніж для попереднього вибору P_0 , оскільки на наступних рівнях алгоритм буде лише збільшувати площу між опорною точкою P_0 та контуром, що будується. Отримані контури для кожної трикутної області об'єднуються в многокутник, що містить усі точки множини S . Далі процедура повторюється для інших опорних точок P_0 . Множина областей усіх опорних точок P_0 , що утворюють різні контури кінчена.

Існування простого многогранника в E^d

Твердження. В просторі E^d для будь якої множини S з $N > d$ точок у загальній позиції, в існує простий многогранник, який включає усі точки множини S .

Зауважимо, що оскільки описаний алгоритм оперує тільки множинними операціями та опуклими оболонками, він а) дозволяє зробити узагальнення на простори довільної розмірності E^d з точністю до заміни фрагментів та суб-фрагментів суміжними гранями-ділянками поверхні многогранника, що будується, та їх об'єднаннями; б) розмічає схему конструктивного доведення існування простого многогранника для множини точок в E^d .

Доведення (схематично). У якості опорної обираємо будь яку з граней $CH(S)$ для наступного кроку рекурсії. Наприклад, як у випадку зі спіральним многокутником, ми можемо обирати першу грань в структурі даних, що містить усі грані поточного фрагменту опуклої оболонки. Для кожного $(k+1)$ -го кроку рекурсії, якщо все ще є «вільні» точки, то вони,

за побудовою на кроці k , є внутрішніми точками опуклої оболонки. При побудові на них опуклої оболонки наступного кроку множини точок перетинаються з нею тільки на опорній грані, яка видаляється з поверхні многогранника, що будується. Тобто отримане об'єднання граней не має самоперетинів і гарантовано завершується додаванням частини поверхні оболонки, що побудована на останньому рівні рекурсії.

Практична реалізація

Практична реалізація основного та допоміжних алгоритмів виконана на мові програмування C++ з мінімальною залежністю від додаткових бібліотек: використано тільки контейнери та алгоритми STL. Програма для наглядної демонстрації роботи алгоритму розроблена для платформи Microsoft Windows з використанням каркасу MFC.

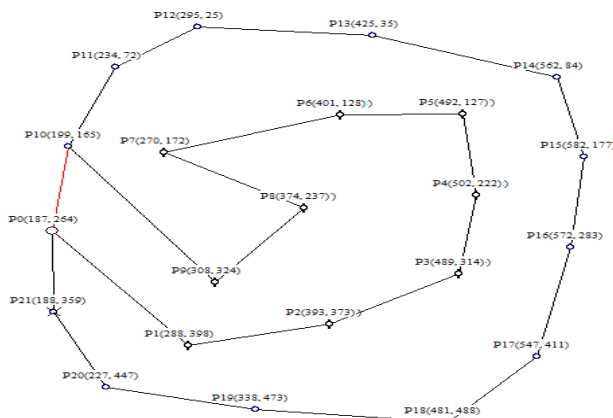


Рис. 2. Практична реалізація алгоритму (приклад з $N=22$ точок).

Висновки

Хоча представлений підхід, як і більшість попередніх відомих евристичних алгоритмів, забезпечує генерацію тільки деякої підмножини многокутників з усіх можливих варіантів, на відміну від них, дозволяє природне узагальнення для просторів довільних вимірів E^d , та може використовуватися, як основа конструктивного доведення існування узагальненої «полігонізації» заданої множини точок для E^d – простого d -вимірного многогранника.

Крім того, деякі конфігурації такі, як спіральні [6, 7], є частинними випадками запропонованого алгоритму при певних значеннях його параметрів. Наведений алгоритм також може служити для збирання точок в ланцюги без самоперетинів.

Спосіб побудови множини опорних точок P_0 залишається відкритим для подальшого доопрацювання та порівняння результатів з [9] і реалізації на основі підходу [10].

Список використаних джерел

1. J. S. B. Mitchell, Joseph O'Rourke, Computational Geometry Column 42, International Journal of Computational Geometry and Applications, Vol. 11, No. 5 (2001), – P.573-582.
2. Branko Grünbaum, Hamiltonian polygons and polyhedra, Geombinatorics, 3, 1994, – P.83-89.
3. Sándor P. Fekete, On simple polygonizations with optimal area, Discrete and Computational Geometry, 23(1), January 2000, – P.73-110.
4. Thomas Auer, Martin Held, Heuristics for the generation of random polygons, Proceedings of the 8th Canadian Conference on Computational Geometry, 1996, – P.38-43.
5. C. Zhu, G. Sundaram, J. Snoeyink, J.S.B. Mitchell, Generating random polygons with given vertices, Computational Geometry: Theory and Applications, Vol. 6, No. 5, 1996, – P.277-290
6. F. Taherkhani¹, A. Nourollah, The generation of pseudo-triangulated spiral polygon using convex hull layers, Proceedings of the Proceedings of the ICFCS, 2011,
7. Justin Iwerks, Joseph S. B. Mitchell, Spiral serpentine polygonization of a planar point set, XIV Spanish Meeting on Computational Geometry, 27–30 June 2011, – P.146-150.
8. L.Mohammadi, A. Nourollah, Generating Sunflower Random Polygons on a Set of Vertices, Proceedings of the ICFCS, 2011, – P.47-50.
9. V. Muravitskiy, V. Tereshchenko, Generating a simple polygonizations, Proceedings of 15th International Conference on Information Visualisation, 2011, – P.502-506.
10. V. N. Tereshchenko, A. V. Anisimov. Recursion and parallel algorithms in geometric modeling problems. *Journal: Cybernetics and Systems Analysis*: Vol. 46, N 2 (2010), – P.173-184.

Надійшла до редколегії 01.10.13

УДК 004.032.26:004.8

Четирбок П.В.¹, аспірант

Распознавание объектов с использованием критерия на основе векторной меры близости образов в пространстве ошибок

В данной работе построено отображение множества образов на множество векторов ошибок распознавания образов нейронной сетью, которое позволяет связать классификацию образов с анализом векторов в пространстве ошибок. Векторный критерий позволяет группировать образы, распознавать, сравнивать и анализировать их.

Сформулирован взвешенный критерий близости образов сигналов в пространстве ошибок. Предложен алгоритм перехода из пространства параметров образов в пространство ошибок распознавания образов.

Построено оптимальное решающее правило для классификации образов сигналов с использованием взвешенного критерия близости распознаваемых образов в пространстве ошибок распознавания.

Достоверность полученных научных результатов, выводов и рекомендаций работы подтверждена результатами экспериментальных исследований разработанной универсальной системы интеллектуального анализа данных, которая решает задачи распознавания объектов электрооптических изображений NEFFClass BGCGG (Basic Gradient Conjugate Gradient Genetic), проведенных на базе «Института прикладного системного анализа» НТТУ «КПИ». Полученные в работе результаты, наглядно демонстрируют эффективность использования разработанных моделей, методов и алгоритмов для решения задач распознавания сигналов.

Ключові слова: критерій, образ, нейронная сеть.

¹Республіканський вищий навчальний заклад «Кримський гуманітарний університет», 986000, м. Ялта, вул. Севастопольська, 2
e-mail: petr58@ukr.net

P.V. Chetyrbok¹, postgraduate

Recognitions of objects with the use of criterion on the basis of vectorial measure of closeness of appearance in space of errors

In the given work displaying is built of great number of appearances on the great number of vectors of errors of pattern recognition by the neuron network, which allows to link classification of appearances with the analysis of vectors in space of errors. A criterion allows to group appearances, recognize, compare them.

A vector criterion is formulated of closeness of signal images in space of errors. An algorithm is offered for transition from space of parameters of signal images in space of errors of pattern recognition.

An optimum decided rule is built for classification of signal images of signals with the use of weighed criterion of closeness of recognizable signal images in space of errors of recognition.

Authenticity of the received scientific results, conclusions and recommendations of this thesis work has been confirmed by the results of experimental researches of the developed universal system of intellectual data analysis, which solve the task of recognition of objects of the electro-optical images NEFFClass BGCGG (Basic Gradient Conjugate Gradient Genetic), conducted on the base of «Institute of the Applied Systems Analysis» NTUU «KPI». The results received in work evidently demonstrate efficiency of the use of the developed models, methods and algorithms for the solution of tasks of recognition of signals.

Key Words: criterion, appearance, neuron network.

¹Crimean University for the Humanities, 986000, Yalta, Sevastopolskaya St., 2
e-mail: petr58@ukr.net

Статтю представив професор, д.ф.-м.н. Буй Д.Б.

© П.В. Четирбок, 2013

Введение. Для нейронных сетей актуальным является модификация методов обучения с использованием в качестве критерия близости образов скалярной величины, равной сумме норм Евклида, обобщенной нормы Евклида и нормы Чебышева. Актуальным является также исследование эффективности модифицированных методов в решении задач с трудно распознаваемыми образами сигналов в условиях помех.

Анализ литературы. Из известных методов обучения нейронных сетей наиболее широкое применение имеют градиентные методы со случайным изменением начальных условий [1]. Недостатком этих методов является трудности распознавания образов в случае близости по норме Евклида сравниваемых образов. Существует много эвристических алгоритмов классификации без учителя [1], основанных на использовании мер близости между объектами. Каждый из них имеет свою область применения, а наиболее распространенным недостатком является отсутствие четкой формализации задачи: совершается переход от идеи кластеризации прямо к алгоритму, в результате неизвестно, что ищется.

1) Базовый способ: для вектора данных x^i и каждого ядра a_i вычисляется $y_i = d(x, a_i)$. По правилу «победитель забирает все» строка ответов y_i преобразуется в строку, где только один элемент, соответствующий максимальному y_i , равен 1, остальные - нули. Эта строка и является результатом функционирования сети. По ней может быть определен номер класса (номер места, на котором стоит 1) и другие показатели.

2) Метод аккредитации: за слоем элементов базового метода, выдающих сигналы 0 или 1 по правилу "победитель забирает все" (далее называем его слоем базового интерпретатора), надстраивается еще один слой выходных сумматоров. С каждым $(i-m)$ классом ассоциируется q -мерный выходной вектор z^i с координатами z_j^i . Он может формироваться по-разному: от двоичного представления номера класса до вектора ядра класса. Вес связи, ведущей от i -го элемента слоя базового интерпретатора к j -му выходному сумматору определяется в точности как z_j^i . Если на этом i -м элементе базового интерпретатора получен сигнал 1, а на остальных - 0, то на выходных сумматорах будут получены числа z_j^i .

3) Нечеткая классификация. Пусть вектор данных x обработан слоем элементов, вычисляющих $y_i = d(x, a_i)$. Идея дальнейшей обработки состоит в том, чтобы выбрать из этого набора $\{y_i\}$ несколько самых больших чисел и после нормировки объявить их значениями функций принадлежности к соответствующим классам. Предполагается, что к остальным классам объект наверняка не принадлежит. Для выбора семейства G наибольших y_i определим следующие числа:

$$y_{\max} = \max \{y_i\}, M_y = \frac{1}{k} \sum_i y_i, s = (1 - \alpha)M_y + \alpha y_{\max},$$

где число α характеризует отклонение "уровня среза" s от среднего значения M_y , $\alpha \in [-1, 1]$, по умолчанию обычно принимается $\alpha = 0$.

Множество $J = \{i | y_i \in G\}$ трактуется как совокупность номеров тех классов, к которым может принадлежать объект, а нормированные на единичную сумму неотрицательные величины

$$f_i = \frac{y_i - s}{\sum_{j \in J} (y_j - s)}$$

(при $i \in J$ и $f = 0$ в противном случае)

интерпретируются как значения функций принадлежности этим классам.

4) Метод интерполяции надстраивается над нечеткой классификацией аналогично тому, как метод аккредитации связан с базовым способом. С каждым классом связывается q -мерный выходной вектор z^i . Строится слой из q выходных сумматоров, каждый из которых должен выдавать свою компоненту выходного вектора. Весовые коэффициенты связей, ведущих от того элемента нечеткого классификатора, который вычисляет f_i , к j -му выходному сумматору определяются как z_j^i . В итоге вектор выходных сигналов сети есть

$$z = \sum_i f_i z^i.$$

В отдельных случаях по смыслу задачи требуется нормировка f_i на единичную сумму квадратов или модулей.

5) Способ построения векторного критерия для распознавания образов. Строится отображение множества распознаваемых образов на множество векторов ошибок распознавания образов нейронную сетью, которое позволяет связать классификацию образов с анализом векторов в пространстве ошибок.

Выбор одного из описанных пяти вариантов использования сети определяется нуждами пользователя. Предлагаемые пять способов покрывают большую часть потребностей. В работе разработан 5 метод.

Постановка проблемы. Построить отображение множества распознаваемых образов (векторов параметров образов) на множество векторов ошибок распознавания образов нейронной сетью, которое позволяет связать классификацию образов с анализом векторов в пространстве ошибок.

Цель статьи – построение функционала и решающего правила (векторного критерия для распознавания образов) для решения задачи классификации образов. Функционал равен скалярному произведению векторов ошибок при распознавании нейронной сетью образов и соответствующих им эталонов. Впервые построено решающее правило для классификации образов в виде утверждения: каждому образу, распознаваемому многослойным персептроном в многофакторном пространстве ошибок будет соответствовать свое значение функционала и образ ближе к эталону, чем больше $\cos(\lambda)$.

$$\cos(\lambda) = \frac{(\bar{E}, \bar{X})}{\|\bar{E}\|_c \|\bar{X}\|_c}$$

, где $\bar{E}$ - вектор ошибок в пространстве ошибок, полученный при распознавании нейронной сетью входного образа, $\bar{X}$ - вектор ошибок, полученный при распознавании нейронной сетью эталона. Что позволяет связать классификацию распознаваемых образов с поведением функционала в многофакторном пространстве ошибок.

Решение проблемы. Формулировка векторного критерия близости образов в пространстве ошибок. Переход из пространства параметров в пространство ошибок. Для распознавания образов построим функционал

$$F = E_2 x_1 + E_3 x_2 + E_4 x_3, \quad (1)$$

E_2 – среднеквадратическая ошибка, полученная при распознавании образа:

$$E_2 = \frac{1}{n} \sqrt{\sum_j^n (y_j - d_j)^2}$$

где y_j – реальное выходное состояние нейрона j выходного слоя нейронной сети при

подаче на ее входы образа; d_j – идеальное (желаемое) выходное состояние этого нейрона. Суммирование ведется по всем нейронам выходного слоя n .

E_3 – линейная ошибка сети, полученная как сумма модулей поразрядного отклонения образа от эталона:

$$E_3 = \frac{1}{n} \sum_j^n |y_j - d_j|$$

где y_j – реальное выходное состояние нейрона j выходного слоя нейронной сети при подаче на ее входы образа; d_j – идеальное (желаемое) выходное состояние этого нейрона. Суммирование ведется по всем нейронам выходного слоя n .

E_4 – максимальная ошибка поразрядного отклонения образа от эталона:

$$E_4 = \max_{j=1 \dots n} |y_j - d_j|$$

где y_j – реальное выходное состояние нейрона j выходного слоя нейронной сети при подаче на ее входы образа; d_j – идеальное (желаемое) выходное состояние этого нейрона. Суммирование ведется по всем нейронам выходного слоя n .

(x_1, x_2, x_3) – вектор ошибок при распознавании эталонного сигнала.

Для каждого из образов получили свое значение функционала, т.е. функционал однозначно определяет образ, запомненный сетью. Если для двух образов величины E_2, E_3, E_4 совпадут, то функционалы для этих образов будут различны. Если в качестве образов использовать волны, то первый член функционала отвечает за отклонение частоты образа от эталонной, второй член отвечает за отклонений по амплитуде волны образа от эталонной, третий член анализирует пиковые всплески. Функционал имеет нижнюю границу при $F=0$. Для образов волновой природы функционал позволяет выделить вредные гармоники, подавить шумы и выделить полезный сигнал.

При распознавании сигнала нейронной сетью вычисляем функционал. При n эталонных образах сигналов мы можем получить n функционалов $F_1, F_2, \dots, F_n$, где n – количество сигналов, которые распознает сеть. Как же определить какому образу соответствует входной сигнал или он не распознается сетью?

Таблица 4.2.6

В соответствии с введенным функционалом (1) минимизируемой целевой функцией ошибки нейронной сети

$$E(w) = \frac{1}{2} \sum_{j,p} (y_{j,p}^{(N)} - d_{j,p})^2$$

является величина:

где $y_{j,p}^{(N)}$ – реальное выходное состояние нейрона j выходного слоя N нейронной сети при подаче на ее входы p -го образа; $d_{j,p}$ – идеальное (желаемое) выходное состояние этого нейрона.

Суммирование ведется по всем нейронам выходного слоя и по всем обрабатываемым сетью образам. Минимизация ведется методом градиентного спуска [1]. При распознавании образа нейронной сетью вычисляем функционал, используя весовые коэффициенты, полученные при обучении сети. Функционал F равен скалярному произведению векторов ошибок (E_2, E_3, E_4) и (x_1, x_2, x_3) .

Найдем косинус угла между векторами ошибок:

$$\cos(\lambda) = \frac{(\vec{E}, \vec{X})}{\|\vec{E}\|_c \|\vec{X}\|_c}$$

Аналогично находим косинусы для всех эталонных образов $F_1, F_2, \dots, F_n$, где n – число эталонных образов предъявленных сети при обучении. Тот функционал F_i , для которого $\cos(\lambda)$ максимален определяет соответствующий входному сигналу эталонный образ. Для расчета взвешенного критерия используется взвешенная норма в пространстве ошибок.

Проведем эксперименты для градиентного алгоритма с использованием взвешенного критерия на основе векторов ошибок при распознавании областей на электрооптических изображениях. Закодируем базу правил с помощью кода Грея (0 - 000, 1 - 001, 2 - 011, 3 - 010, 4 - 110).

Используя 54 образов (по 6 с каждого класса) научим двухслойный персептрон распознавать образцы согласно базе правил. Остальные 45 образцов будут использованы для тестирования.

$$F = (\vec{E}, \vec{X}) = \cos(\lambda) \|\vec{E}\|_c \|\vec{X}\|_c$$

База правил

№ правила	Код признаков	№ Класс
1	1 110 110 110 110	0
2	1 110 000 001 110	1
3	1 110 000 000 110	1
4	1 110 001 000 110	1
5	1 011 010 001 001	2
6	1 001 000 001 000	3
7	1 110 110 001 110	4
8	1 010 110 010 010	5
9	1 010 010 011 010	5
10	1 110 110 010 010	5
11	1 000 000 000 000	6
12	1 010 011 001 011	7
13	1 001 000 000 001	8
14	1 001 001 000 001	8
15	1 001 000 000 000	8

Проведены эксперименты на реальных данных и установлено, что использование нейро-нечеткого классификатора NEFClass целесообразно применять для распознавания областей на электрооптических изображениях. При тестировании были ошибочно классифицированы 4 образца с проверочного набора данных [2]. Но для градиентного метода с использованием взвешенного критерия на основе векторов ошибок уровень правильной классификации на проверочной части выборки выше, а именно ошибочно классифицирован 1 образец с проверочного набора данных (Табл. 4.2.7.).

Таблица 4.2.7

Сравнительная таблица для алгоритмов обучения весовых коэффициентов

Алгоритм обуч. весовых коэфф.	Обучение		Тестирование	
	Ср.кв.др. ошибка	Ошибка класс-ии	Среднекв.др. ошибка	Ошибка класс.-ии
Классический	6,650668	0	7,285827	4
Градиентный	5,9893	0	6,829068	4
Сопр. градиента	1,132871	0	3,314763	4
Генетический с треугольной ФП	11,110936	0	13,677424	4
Генетический с гауссовской ФП	3,204446	0	4,568338	4
Градиентный с исп. взвешен. критерия		0		1

Выводы. Каждому образу, распознаваемому многослойным персептроном в многофакторном пространстве ошибок соответствует свой вектор ошибок. Впервые построено решающее правило для классификации образов в виде утверждения: каждому образу, распознаваемому многослойным персептроном в многофакторном пространстве ошибок будет соответствовать свой вектор ошибок и образ ближе к эталону, чем больше $\cos(\lambda)$.

$$\cos(\lambda) = \frac{(\overline{E}, \overline{X})}{\|\overline{E}\|_c \|\overline{X}\|_c},$$

где E - вектор ошибок в пространстве ошибок, полученный при распознавании нейронной сетью входного образа, X - вектор ошибок, полученный при распознавании нейронной сетью эталона. Предложенный в статье функционал и решающее правило (взвешенный критерий на основе векторов ошибок для распознавания образов) позволяют улучшить распознавания областей на электрооптических изображениях.

Список використаних джерел

1. Haykin, Simon Neural Networks: A Comprehensive Foundation Second Edition. Prentice Hall 2006. – 1104 p. (in Russian)
2. Zaychenko, Y.P. Fuzzy models and methods in intelligent systems. – K.: Publishing house «Word», 2008. – 344 p. (in Russian)

Надійшла до редколегії 18.06.13

УДК 004.42:510.69

Шкільняк О.С., к. ф.-м. н.

Секвенційні числення логік часткових предикатів з розширеними реномінаціями

Досліджено нові класи першопорядкових композиційно-номінативних логік часткових однозначних та часткових неоднозначних предикатів. Особливістю цих логік є використання композицій розширеної реномінації. Розглянуто властивості відношень логічного наслідку для множин формул. На цій основі для таких логік запропоновано числення секвенційного типу. Доведено теореми коректності та повноти цих числень.

Ключові слова: логіка, предикат, реномінація, логічний наслідок, секвенційне числення.

Київський національний університет імені Тараса Шевченка, пр. Глушкова 4д, м. Київ, 03680
e-mail: me.oksana@gmail.com

Статтю представив д. ф.-м. н., проф. Буй Д.Б.

Розвиток інформаційних технологій та програмування робить актуальною проблему побудови нових, програмно-орієнтованих логічних формалізмів. Низку різноманітних логічних систем розроблено (див., напр., [1, 2]) на базі спільного для логіки й програмування композиційно-номінативного підходу. Логіки, збудовані на цій основі, названо композиційно-номінативними (КНЛ).

Нові класи першопорядкових КНЛ часткових предикатів запропоновано в [3]. Характерною їх особливістю є використання композицій розширеної реномінації (перейменування). Це дає змогу явно задавати відсутність значення для предметних імен (змінних). Запропоновані чисті першопорядкові КНЛ з розширеними реномінаціями (ЧКНЛРР) утворюють підрівень чистих першопорядкових КНЛ (ЧКНЛ). Семантичні властивості ЧКНЛРР далі досліджені в [4], для формалізації традиційного відношення логічного наслідку в ЧКНЛРР часткових однозначних предикатів там запропоновано числення секвенційного типу.

Дана стаття продовжує дослідження, розпочаті в [3, 4]. Основною її метою є побудова числень секвенційного типу для різних відношень логічного наслідку в ЧКНЛРР часткових однозначних та часткових неоднозначних предикатів. Для побудованих числень доведено теореми коректності та повноти.

O.S. Shkilniak, PhD (Phys.-Math.).

Sequent calculi of logics of partial predicates with extended renominations

We study new classes of first-order composition-nominative logics of partial single-valued and partial multi-valued predicates. Using of extended renomination composition is a special feature of the introduced logics. Properties of relation of logical consequence for sets of formulas are described. Sequent calculi based on such logics are specified, and their soundness and completeness are proved.

Key words: logic, predicate, renomination, logical consequence, sequent calculus.

Taras Shevchenko National University of Kyiv, 4d Glushkova str., Kyiv, 03680
e-mail: me.oksana@gmail.com

Поняття, які тут не визначаються, тлумачимо в сенсі робіт [1–3, 5]. Дотримуємось позначень робіт [3, 4].

1. ЧКНЛ з розширеними реномінаціями

Композиції розширеної реномінації для квазіарних предикатів задаємо за допомогою операцій розширеної реномінації для іменних множин.

Для іменних множин (ІМ) операцію розширеної реномінації $r_{x_1, \dots, x_n, \perp, \dots, \perp}^{v_1, \dots, v_n, u_1, \dots, u_m} : {}^V A \rightarrow {}^V A$ задамо так:

$$r_{x_1, \dots, x_n, \perp, \dots, \perp}^{v_1, \dots, v_n, u_1, \dots, u_m}(\delta) = \\ = \delta \parallel_{\{v_1, \dots, v_n, u_1, \dots, u_m\}} + [v_1 \mapsto \delta(x_1), \dots, v_n \mapsto \delta(x_n)].$$

Тут ${}^V A$ – клас всіх V -ІМ над A , тобто однозначних функцій вигляду $\delta : V \rightarrow A$, де V і A – множини предметних імен і предметних значень.

Спеціальний символ $\perp$ вказує на відсутність значення для відповідного верхнього імені.

Ввівши позначення $\bar{u}$ для $u_1, \dots, u_n$, скорочено пишемо $r_{\bar{x}, \perp}^{\bar{v}, \bar{u}}$ замість $r_{x_1, \dots, x_n, \perp, \dots, \perp}^{v_1, \dots, v_n, u_1, \dots, u_m}$.

Довільну функцію вигляду $P : {}^V A \rightarrow \{T, F\}$ будемо називати V -квазіарним предикатом на A

Область істинності та область хибності предиката $P : {}^V A \rightarrow \{T, F\}$ – це множини

$$T(P) = \{d \in {}^V A \mid T \in P(d)\} \text{ та} \\ F(P) = \{d \in {}^V A \mid F \in P(d)\}.$$

Якщо P однозначний, то $T(P) \cap F(P) = \emptyset$.

Якщо P тотальний, то $T(P) \cup F(P) = {}^V A$.

Композицію розширеної реномінації $R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}$ задаємо так. Для довільних $P \in Pr^A$ та $d \in {}^V A$ маємо

$$R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}(P)(d) = P(r_{\bar{x}, \perp}^{\bar{v}, \bar{u}}(d)).$$

Тут Pr^A – клас V -квазіарних предикатів на A .

Властивості композицій $R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}$ аналогічні відповідним властивостям композицій $R_{\bar{x}}^{\bar{v}}$ (див. [3]).

Для ЧКНЛРР маємо [4] такі співвідношення:

$$T(R_{\bar{v}, \perp, z}^{\bar{u}, \bar{w}, x}(P)) \cap F(\varepsilon z) \subseteq T(R_{\bar{v}, \perp}^{\bar{u}, \bar{w}}(\exists x P)); \quad (TR_{\perp} \exists R)$$

$$F(R_{\bar{v}, \perp}^{\bar{u}, \bar{w}}(\exists x P)) \cap F(\varepsilon z) \subseteq F(R_{\bar{v}, \perp, z}^{\bar{u}, \bar{w}, x}(P)). \quad (FR_{\perp} \exists R)$$

Тут εz – параметризовані за предметними іменами предикати-індикатори εz , які визначають наявність в даних компоненти з відповідним z .

Предикати-індикатори εz задаємо [3] так:

$$T(\varepsilon z) = \{d \mid d(z) \uparrow\} = \{d \in {}^V A \mid z \notin \text{asn}(d)\};$$

$$F(\varepsilon z) = \{d \mid d(z) \downarrow\} = \{d \in {}^V A \mid z \in \text{asn}(d)\}.$$

Мова та семантичні моделі ЧКНЛРР описуються аналогічно випадку традиційних ЧКНЛ.

Властивості формул ЧКНЛРР пов'язані з композиціями розширеної реномінації, описано в [3]. Зосередимося далі на властивостях відношень логічного наслідку для множин формул. Для відношення $\models_{Cl}$ такі властивості описано в [4].

Відношення наслідку для множин формул при інтерпретації на фіксованій моделі мови A задаємо [2] так. Нехай Γ та Δ – множини формул

$$\Gamma_A \models_{Cl} \Delta, \text{ якщо } \bigcap_{\Phi \in \Gamma} T(\Phi_A) \cap \bigcap_{\Psi \in \Delta} F(\Psi_A) = \emptyset;$$

$$\Gamma_A \models_T \Delta, \text{ якщо } \bigcap_{\Phi \in \Gamma} T(\Phi_A) \subseteq \bigcup_{\Psi \in \Delta} T(\Psi_A);$$

$$\Gamma_A \models_F \Delta, \text{ якщо } \bigcap_{\Psi \in \Delta} F(\Psi_A) \subseteq \bigcup_{\Phi \in \Gamma} F(\Phi_A);$$

$$\Gamma_A \models_{TF} \Delta, \text{ якщо } \Gamma_A \models_T \Delta \text{ та } \Gamma_A \models_F \Delta.$$

Відношення логічного наслідку для множин формул $\models_{Cl}, \models_T, \models_F, \models_{TF}$ задаємо за схемою:

$$\Gamma \models_* \Delta, \text{ якщо } \Gamma_A \models_* \Delta \text{ для кожної моделі мови } A.$$

Надалі, якщо інше не зазначено, $\models$ позначає:

– одне із $\models_{Cl}, \models_T, \models_F, \models_{TF}$ для неокласичної семантики;

– $\models_{TF}$ для загальної семантики.

2. Властивості відношень логічного наслідку для множин формул

Наведемо основні властивості відношень логічного наслідку для множин формул ЧКНЛРР. Від властивостей відношень логічного наслідку для множин формул ЧКНЛ вони відрізняються лише використанням символів $R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}$ замість $R_{\bar{x}}^{\bar{v}}$.

С) Якщо $\Gamma \cap \Delta \neq \emptyset$, то $\Gamma \models \Delta$.

У) Нехай $\Gamma \subseteq \Lambda$ та $\Delta \subseteq \Sigma$, тоді $\Gamma \models \Delta \Rightarrow \Lambda \models \Sigma$.

Властивість C гарантує наявність кожного з логічних наслідків $\models_{Cl}, \models_T, \models_F, \models_{TF}$.

Додатково гарантують наявність відповідного логічного наслідку властивості:

$$(CL) \Phi, \neg \Phi, \Gamma \models \Delta$$

($\models$ – це $\models_T$ або $\models_{Cl}$, неокласична семантика);

$$(CR) \Gamma \models \Delta, \Phi, \neg \Phi$$

($\models$ – це $\models_F$ або $\models_{Cl}$, неокласична семантика);

$$(CLR) \Phi, \neg \Phi, \Gamma \models_{TF} \Delta, \Psi, \neg \Psi.$$

Властивості CL, CR, CLR істотні лише для відношень $\models_T, \models_F, \models_{TF}$; для $\models_{Cl}$ вони є похідними.

Базові властивості пропозиційного рівня:

$$\neg \neg \vdash \vdash) \neg \neg \Phi, \Gamma \models \Delta \Leftrightarrow \Phi, \Gamma \models \Delta;$$

$$\neg \neg \vdash \vdash) \Gamma \models \Delta, \neg \neg \Phi \Leftrightarrow \Gamma \models \Delta, \Phi.$$

$$\vee \vdash \vdash) \Phi \vee \Psi, \Gamma \models \Delta \Leftrightarrow \Phi, \Gamma \models \Delta \text{ та } \Psi, \Gamma \models \Delta;$$

$$\vee \vdash \vdash) \Gamma \models \Delta, \Phi \vee \Psi \Leftrightarrow \Gamma \models \Delta, \Phi, \Psi.$$

$$\neg \vee \vdash \vdash) \neg(\Phi \vee \Psi), \Gamma \models \Delta \Leftrightarrow \neg \Phi, \neg \Psi, \Gamma \models \Delta;$$

$$\neg \vee \vdash \vdash) \Gamma \models \Delta, \neg(\Phi \vee \Psi) \Leftrightarrow \Gamma \models \Delta, \neg \Phi \text{ та } \Gamma \models \Delta, \neg \Psi.$$

Для $\models_{Cl}$ також додатково маємо:

$$\neg \vdash \vdash) \neg \Phi, \Gamma \models \Delta \Leftrightarrow \Gamma \models \Delta, \Phi;$$

$$\neg \vdash \vdash) \Gamma \models \Delta, \neg \Phi \Leftrightarrow \Phi, \Gamma \models \Delta.$$

Теорема 1 (заміни еквівалентних). Нехай $\Phi \sim_{TF} \Psi$. Тоді:

$$\Phi, \Gamma \models \Delta \Leftrightarrow \Psi, \Gamma \models \Delta \text{ та } \Gamma \models \Delta, \Phi \Leftrightarrow \Gamma \models \Delta, \Psi.$$

Наведені нижче властивості індуковані відповідними властивостями композицій $R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}$ із використанням теореми заміни еквівалентних.

$$R_{\perp} T_{\perp}) R_{z, \bar{x}, \perp}^{z, \bar{v}, \bar{u}}(\Phi), \Gamma \models \Delta \Leftrightarrow R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}(\Phi), \Gamma \models \Delta;$$

$$R_{\perp} T_{\perp}) \Gamma \models \Delta, R_{z, \bar{x}, \perp}^{z, \bar{v}, \bar{u}}(\Phi) \Leftrightarrow \Gamma \models \Delta, R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}(\Phi);$$

$$\neg R_{\perp} T_{\perp}) \neg R_{z, \bar{x}, \perp}^{z, \bar{v}, \bar{u}}(\Phi), \Gamma \models \Delta \Leftrightarrow \neg R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}(\Phi), \Gamma \models \Delta;$$

$$\neg R_{\perp} T_{\perp}) \Gamma \models \Delta, \neg R_{z, \bar{x}, \perp}^{z, \bar{v}, \bar{u}}(\Phi) \Leftrightarrow \Gamma \models \Delta, \neg R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}(\Phi);$$

$$\Phi_{\perp} N_{\perp}) R_{z, \bar{x}, \perp}^{y, \bar{v}, \bar{u}}(\Phi), \Gamma \models \Delta \Leftrightarrow R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}(\Phi), \Gamma \models \Delta;$$

$$\Phi_{\perp} N_{\perp}) \Gamma \models \Delta, R_{z, \bar{x}, \perp}^{y, \bar{v}, \bar{u}}(\Phi) \Leftrightarrow \Gamma \models \Delta, R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}(\Phi).$$

$$\neg \Phi_{\perp} N_{\perp}) \neg R_{z, \bar{x}, \perp}^{y, \bar{v}, \bar{u}}(\Phi), \Gamma \models \Delta \Leftrightarrow \neg R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}(\Phi), \Gamma \models \Delta;$$

$$\neg \Phi_{\perp} N_{\perp}) \Gamma \models \Delta, \neg R_{z, \bar{x}, \perp}^{y, \bar{v}, \bar{u}}(\Phi) \Leftrightarrow \Gamma \models \Delta, \neg R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}(\Phi).$$

Для властивостей типу $\Phi_{\perp} N$ та $\neg \Phi_{\perp} N$: $y \in v(\Phi)$.

$$R_{\perp} R_{\perp}) R_{\bar{y}, \perp}^{\bar{v}, \bar{z}}(R_{\bar{x}, \perp}^{\bar{u}, \bar{t}}(\Phi)), \Gamma \models \Delta \Leftrightarrow R_{\bar{y}, \perp}^{\bar{v}, \bar{z}} \circ R_{\bar{x}, \perp}^{\bar{u}, \bar{t}}(\Phi), \Gamma \models \Delta;$$

$$R_{\perp} R_{\perp}) \Gamma \models \Delta, R_{\bar{y}, \perp}^{\bar{v}, \bar{z}}(R_{\bar{x}, \perp}^{\bar{u}, \bar{t}}(\Phi)) \Leftrightarrow \Gamma \models \Delta, R_{\bar{y}, \perp}^{\bar{v}, \bar{z}} \circ R_{\bar{x}, \perp}^{\bar{u}, \bar{t}}(\Phi);$$

$$\neg R_{\perp} R_{\perp}) \neg R_{\bar{y}, \perp}^{\bar{v}, \bar{z}}(R_{\bar{x}, \perp}^{\bar{u}, \bar{t}}(\Phi)), \Gamma \models \Delta \Leftrightarrow$$

$$\Leftrightarrow \neg R_{\bar{y}, \perp}^{\bar{v}, \bar{z}} \circ R_{\bar{x}, \perp}^{\bar{u}, \bar{t}}(\Phi), \Gamma \models \Delta;$$

$$\neg R_{\perp} R_{\perp}) \Gamma \models \Delta, \neg R_{\bar{y}, \perp}^{\bar{v}, \bar{z}} (R_{\bar{x}, \perp}^{\bar{u}, \bar{t}} (\Phi)) \Leftrightarrow$$

$$\Leftrightarrow \Gamma \models \Delta, \neg R_{\bar{y}, \perp}^{\bar{v}, \bar{z}} \circ_{\bar{x}, \perp}^{\bar{u}, \bar{t}} (\Phi);$$

$$R_{\perp} \neg \vdash) R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\neg \Phi), \Gamma \models \Delta \Leftrightarrow \neg R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Phi), \Gamma \models \Delta;$$

$$R_{\perp} \neg \vdash) \Gamma \models \Delta, R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\neg \Phi) \Leftrightarrow \Gamma \models \Delta, \neg R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Phi);$$

$$\neg R_{\perp} \neg \vdash) \neg R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\neg \Phi), \Gamma \models \Delta \Leftrightarrow R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Phi), \Gamma \models \Delta;$$

$$\neg R_{\perp} \neg \vdash) \Gamma \models \Delta, \neg R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\neg \Phi) \Leftrightarrow \Gamma \models \Delta, R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Phi);$$

$$R_{\perp} \vee \vdash) R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Phi \vee \Psi), \Gamma \models \Delta \Leftrightarrow$$

$$\Leftrightarrow R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Phi) \vee R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Psi), \Gamma \models \Delta;$$

$$R_{\perp} \vee \vdash) \Gamma \models \Delta, R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Phi \vee \Psi) \Leftrightarrow$$

$$\Leftrightarrow \Gamma \models \Delta, R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Phi) \vee R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Psi);$$

$$\neg R_{\perp} \vee \vdash) \neg R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Phi \vee \Psi), \Gamma \models \Delta \Leftrightarrow$$

$$\Leftrightarrow \neg R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Phi), \neg R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Psi), \Gamma \models \Delta;$$

$$\neg R_{\perp} \vee \vdash) \Gamma \models \Delta, \neg R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Phi \vee \Psi) \Leftrightarrow$$

$$\Leftrightarrow \Gamma \models \Delta, \neg R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Phi) \text{ та } \Gamma \models \Delta, \neg R_{\bar{x}, \perp}^{\bar{v}, \bar{u}} (\Psi);$$

$$R_{\perp} \exists R_{\perp}) R_{\bar{v}, \perp}^{\bar{u}, \bar{w}, x} (\exists x \Phi), \Gamma \models \Delta \Leftrightarrow R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Gamma \models \Delta;$$

$$R_{\perp} \exists R_{\perp}) \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}, x} (\exists x \Phi) \Leftrightarrow \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi);$$

$$\neg R_{\perp} \exists R_{\perp}) R_{\bar{v}, \perp}^{\bar{u}, \bar{w}, x} (\neg \exists x \Phi), \Gamma \models \Delta \Leftrightarrow$$

$$\Leftrightarrow \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Gamma \models \Delta;$$

$$\neg R_{\perp} \exists R_{\perp}) \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}, x} (\neg \exists x \Phi) \Leftrightarrow$$

$$\Leftrightarrow \Gamma \models \Delta, \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi);$$

$$R_{\perp} \exists p_{\perp}) R_y^x (\exists x \Phi), \Gamma \models \Delta \Leftrightarrow \exists x \Phi, \Gamma \models \Delta;$$

$$R_{\perp} \exists p_{\perp}) \Gamma \models \Delta, R_y^x (\exists x \Phi) \Leftrightarrow \Gamma \models \Delta, \exists x \Phi;$$

$$\neg R_{\perp} \exists p_{\perp}) R_y^x (\neg \exists x \Phi), \Gamma \models \Delta \Leftrightarrow \neg \exists x \Phi, \Gamma \models \Delta;$$

$$\neg R_{\perp} \exists p_{\perp}) \Gamma \models \Delta, R_y^x (\neg \exists x \Phi) \Leftrightarrow \Gamma \models \Delta, \neg \exists x \Phi.$$

Для властивостей типу $R_{\perp} \exists R$, $\neg R_{\perp} \exists R$ та $R_{\perp} \exists p$, $\neg R_{\perp} \exists p$ маємо: $y \in V \cup \{\perp\}$.

Для властивостей типу $R_{\perp} \exists R$ та $\neg R_{\perp} \exists R$ неявно маємо умову $x \notin \{\bar{u}, \bar{w}\}$.

Для опису властивостей елімінації кванторів використаємо предикати-індикатори εz .

$$\exists R_{\perp} \vdash) R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Gamma \models \Delta \Leftrightarrow R_{\bar{v}, \perp, z}^{\bar{u}, \bar{w}, x} (\Phi), \Gamma \models \Delta, \varepsilon z$$

(за умови $z \in V_T$ та $z \notin nm(\Gamma, \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi))$).

$$\exists \vdash) \exists x \Phi, \Gamma \models \Delta \Leftrightarrow R_z^x (\Phi), \Gamma \models \Delta, \varepsilon z$$

(за умови $z \in V_T$ та $z \notin nm(\Gamma, \Delta, \exists x \Phi)$).

$$\neg \exists R_{\perp} \vdash) \Gamma \models \Delta, \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi) \Leftrightarrow$$

$$\Leftrightarrow \Gamma \models \Delta, \neg R_{\bar{v}, \perp, z}^{\bar{u}, \bar{w}, x} (\Phi), \Gamma \models \Delta, \varepsilon z$$

(за умови $z \in V_T$ та $z \notin nm(\Gamma, \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi))$).

$$\exists f_{\perp} \vdash) \Gamma \models \Delta, \exists x \Phi \Leftrightarrow \Gamma \models \Delta, \exists x \Phi, R_z^x (\Phi), \varepsilon z$$

(за умови $z \in V_T$ та $z \notin nm(\Gamma, \Delta, \exists x \Phi)$).

$$\neg \exists R_{\perp} f_{\perp} \vdash) \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Gamma \models \Delta \Leftrightarrow$$

$$\Leftrightarrow \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \neg R_{\bar{v}, \perp, z}^{\bar{u}, \bar{w}, x} (\Phi), \Gamma \models \Delta, \varepsilon z$$

(за умови $z \in V_T$ та $z \notin nm(\Gamma, \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi))$).

$$\neg \exists f_{\perp} \vdash) \neg \exists x \Phi, \Gamma \models \Delta \Leftrightarrow \neg \exists x \Phi, \neg R_z^x (\Phi), \Gamma \models \Delta, \varepsilon z$$

(за умови $z \in V_T$ та $z \notin nm(\Gamma, \Delta, \exists x \Phi)$).

Властивості $\exists f_{\perp}$, $\neg \exists f_{\perp}$ та $\exists R_{\perp} f_{\perp}$, $\neg \exists R_{\perp} f_{\perp}$ назвемо властивостями типу $\exists f$ ($\exists$ -first).

$$\exists R_{\perp} \vee \vdash) \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \varepsilon y \Leftrightarrow$$

$$\Leftrightarrow \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), R_{\bar{v}, \perp, y}^{\bar{u}, \bar{w}, x} (\Phi), \varepsilon y.$$

$$\exists \vee \vdash) \Gamma \models \Delta, \exists x \Phi, \varepsilon y \Leftrightarrow \Gamma \models \Delta, \exists x \Phi, R_y^x (\Phi), \varepsilon y.$$

$$\neg \exists R_{\perp} \vee \vdash) \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Gamma \models \Delta, \varepsilon y \Leftrightarrow$$

$$\Leftrightarrow \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \neg R_{\bar{v}, \perp, y}^{\bar{u}, \bar{w}, x} (\Phi), \Gamma \models \Delta, \varepsilon y.$$

$$\neg \exists \vee \vdash) \neg \exists x \Phi, \Gamma \models \Delta, \varepsilon y \Leftrightarrow \neg \exists x \Phi, \neg R_y^x (\Phi), \Gamma \models \Delta, \varepsilon y.$$

Властивості $\exists \vee$, $\neg \exists \vee$ та $\exists R_{\perp} \vee$, $\neg \exists R_{\perp} \vee$ назвемо властивостями типу $\exists \vee$ ($\exists$ -valued).

$$\exists R_{\perp} d_{\perp} \vdash) \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi) \Leftrightarrow$$

$$\Leftrightarrow \varepsilon y, \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi) \text{ та } \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), R_{\bar{v}, \perp, y}^{\bar{u}, \bar{w}, x} (\Phi), \varepsilon y.$$

$$\exists d_{\perp} \vdash) \Gamma \models \Delta, \exists x \Phi \Leftrightarrow \varepsilon y, \Gamma \models \Delta, \exists x \Phi \text{ та } \Gamma \models \Delta, \exists x \Phi, R_y^x (\Phi), \varepsilon y.$$

$$\neg \exists R_{\perp} d_{\perp} \vdash) \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Gamma \models \Delta \Leftrightarrow$$

$$\Leftrightarrow \varepsilon y, \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Gamma \models \Delta \text{ та } \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \neg R_{\bar{v}, \perp, y}^{\bar{u}, \bar{w}, x} (\Phi), \Gamma \models \Delta, \varepsilon y.$$

$$\neg \exists d_{\perp} \vdash) \neg \exists x \Phi, \Gamma \models \Delta \Leftrightarrow \varepsilon y, \neg \exists x \Phi, \Gamma \models \Delta \text{ та } \neg \exists x \Phi, \neg R_y^x (\Phi), \Gamma \models \Delta, \varepsilon y.$$

Властивості $\exists d_{\perp}$, $\neg \exists d_{\perp}$ та $\exists R_{\perp} d_{\perp}$, $\neg \exists R_{\perp} d_{\perp}$ будемо називати властивостями типу $\exists d$.

$$\neg \exists \vdash) \Gamma \models \Delta, \neg \exists x \Phi \Leftrightarrow \Gamma \models \Delta, \neg R_z^x (\Phi), \varepsilon z$$

(за умови $z \in V_T$ та $z \notin nm(\Gamma, \Delta, \exists x \Phi)$).

$$\exists R_{\perp} f_{\perp} \vdash) \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi) \Leftrightarrow$$

$$\Leftrightarrow \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), R_{\bar{v}, \perp, z}^{\bar{u}, \bar{w}, x} (\Phi), \varepsilon z$$

(за умови $z \in V_T$ та $z \notin nm(\Gamma, \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi))$).

$$\exists f_{\perp} \vdash) \Gamma \models \Delta, \exists x \Phi \Leftrightarrow \Gamma \models \Delta, \exists x \Phi, R_z^x (\Phi), \varepsilon z$$

(за умови $z \in V_T$ та $z \notin nm(\Gamma, \Delta, \exists x \Phi)$).

$$\neg \exists R_{\perp} f_{\perp} \vdash) \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Gamma \models \Delta \Leftrightarrow$$

$$\Leftrightarrow \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \neg R_{\bar{v}, \perp, z}^{\bar{u}, \bar{w}, x} (\Phi), \Gamma \models \Delta, \varepsilon z$$

(за умови $z \in V_T$ та $z \notin nm(\Gamma, \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi))$).

$$\neg \exists f_{\perp} \vdash) \neg \exists x \Phi, \Gamma \models \Delta \Leftrightarrow \neg \exists x \Phi, \neg R_z^x (\Phi), \Gamma \models \Delta, \varepsilon z$$

(за умови $z \in V_T$ та $z \notin nm(\Gamma, \Delta, \exists x \Phi)$).

Властивості $\exists f_{\perp}$, $\neg \exists f_{\perp}$ та $\exists R_{\perp} f_{\perp}$, $\neg \exists R_{\perp} f_{\perp}$ назвемо властивостями типу $\exists f$ ($\exists$ -first).

$$\exists R_{\perp} \vee \vdash) \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \varepsilon y \Leftrightarrow$$

$$\Leftrightarrow \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), R_{\bar{v}, \perp, y}^{\bar{u}, \bar{w}, x} (\Phi), \varepsilon y.$$

$$\exists \vee \vdash) \Gamma \models \Delta, \exists x \Phi, \varepsilon y \Leftrightarrow \Gamma \models \Delta, \exists x \Phi, R_y^x (\Phi), \varepsilon y.$$

$$\neg \exists R_{\perp} \vee \vdash) \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Gamma \models \Delta, \varepsilon y \Leftrightarrow$$

$$\Leftrightarrow \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \neg R_{\bar{v}, \perp, y}^{\bar{u}, \bar{w}, x} (\Phi), \Gamma \models \Delta, \varepsilon y.$$

$$\neg \exists \vee \vdash) \neg \exists x \Phi, \Gamma \models \Delta, \varepsilon y \Leftrightarrow \neg \exists x \Phi, \neg R_y^x (\Phi), \Gamma \models \Delta, \varepsilon y.$$

Властивості $\exists \vee$, $\neg \exists \vee$ та $\exists R_{\perp} \vee$, $\neg \exists R_{\perp} \vee$ назвемо властивостями типу $\exists \vee$ ($\exists$ -valued).

$$\exists R_{\perp} d_{\perp} \vdash) \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi) \Leftrightarrow$$

$$\Leftrightarrow \varepsilon y, \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi) \text{ та } \Gamma \models \Delta, R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), R_{\bar{v}, \perp, y}^{\bar{u}, \bar{w}, x} (\Phi), \varepsilon y.$$

$$\exists d_{\perp} \vdash) \Gamma \models \Delta, \exists x \Phi \Leftrightarrow \varepsilon y, \Gamma \models \Delta, \exists x \Phi \text{ та } \Gamma \models \Delta, \exists x \Phi, R_y^x (\Phi), \varepsilon y.$$

$$\neg \exists R_{\perp} d_{\perp} \vdash) \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Gamma \models \Delta \Leftrightarrow$$

$$\Leftrightarrow \varepsilon y, \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Gamma \models \Delta \text{ та } \neg R_{\bar{v}, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \neg R_{\bar{v}, \perp, y}^{\bar{u}, \bar{w}, x} (\Phi), \Gamma \models \Delta, \varepsilon y.$$

$$\neg \exists d_{\perp} \vdash) \neg \exists x \Phi, \Gamma \models \Delta \Leftrightarrow \varepsilon y, \neg \exists x \Phi, \Gamma \models \Delta \text{ та } \neg \exists x \Phi, \neg R_y^x (\Phi), \Gamma \models \Delta, \varepsilon y.$$

3. Секвенційні числення ЧКНЛРР

Властивості відношень логічного наслідку для множин формул ЧКНЛРР є семантичною основою побудови для таких логік числень секвенційного типу.

Секвенційне числення задається базовими секвенційними формами та умовами замкненості секвенції (умови замкненості див. [5]).

Розглянемо різновидності секвенційних числень ЧКНЛРР часткових предикатів.

Q_1G -числення формалізує відношення $\models_{TF}$ для випадку загальної семантики.

Умова замкненості секвенції: $C \vee UnC$.

Базові секвенційні форми Q_1G -числень:

$\vdash R_1T, \vdash R_1T, \vdash \neg R_1T, \vdash \neg R_1T, \vdash \Phi_1N, \vdash \Phi_1N, \vdash \neg \Phi_1N,$
 $\vdash \neg \Phi_1N, \vdash R_1\exists p, \vdash R_1\exists p, \vdash \neg R_1\exists p, \vdash \neg R_1\exists p, \vdash R_1\exists R,$
 $\vdash R_1\exists R, \vdash \neg R_1\exists R, \vdash \neg R_1\exists R, \vdash \neg \neg, \vdash \neg \neg, \vdash \vee, \vdash \vee, \vdash \neg \vee,$
 $\vdash \neg \vee, \vdash R_1R, \vdash R_1R, \vdash \neg R_1R, \vdash \neg R_1R, \vdash R_1\neg, \vdash R_1\neg,$
 $\vdash \neg R_1\neg, \vdash \neg R_1\neg, \vdash R_1\vee, \vdash R_1\vee, \vdash \neg R_1\vee, \vdash \neg R_1\vee, \vdash \exists \epsilon,$
 $\vdash \neg \exists \epsilon, \vdash \exists R_1\epsilon, \vdash \neg \exists R_1\epsilon, \vdash \exists f, \vdash \neg \exists f, \vdash \exists R_1f, \vdash \neg \exists R_1f, \vdash \exists v,$
 $\vdash \neg \exists v, \vdash \exists R_1v, \vdash \neg \exists R_1v, \vdash \exists d, \vdash \neg \exists d, \vdash \exists R_1d, \vdash \neg \exists R_1d.$

Форми $\vdash \neg \neg, \vdash \neg \neg, \vdash \vee, \vdash \vee, \vdash \neg \vee, \vdash \neg \vee, \vdash \exists \epsilon,$
 $\vdash \neg \exists \epsilon, \vdash \exists f, \vdash \neg \exists f, \vdash \exists v, \vdash \neg \exists v, \vdash \exists d, \vdash \neg \exists d$ такі ж, як для відповідних QSG-числень ЧКНЛ (див. [5]).

Форми $\vdash R_1T, \vdash R_1T, \vdash \neg R_1T, \vdash \neg R_1T, \vdash \Phi_1N, \vdash \Phi_1N,$
 $\vdash \neg \Phi_1N, \vdash \neg \Phi_1N, \vdash R_1\exists p, \vdash R_1\exists p, \vdash \neg R_1\exists p, \vdash \neg R_1\exists p,$
 $\vdash R_1\exists R, \vdash R_1\exists R, \vdash \neg R_1\exists R, \vdash \neg R_1\exists R, \vdash R_1R, \vdash R_1R,$
 $\vdash \neg R_1R, \vdash \neg R_1R, \vdash R_1\neg, \vdash R_1\neg, \vdash \neg R_1\neg, \vdash \neg R_1\neg, \vdash R_1\vee,$
 $\vdash R_1\vee, \vdash \neg R_1\vee, \vdash \neg R_1\vee, \vdash \exists R_1\epsilon, \vdash \neg \exists R_1\epsilon, \vdash \exists R_1f, \vdash \neg \exists R_1f,$
 $\vdash \exists R_1v, \vdash \neg \exists R_1v, \vdash \exists R_1d, \vdash \neg \exists R_1d$ відрізняються від відповідних форм числення QSG лише наявністю розширених реномінацій.

Наведемо базові форми числення Q_1G :

$$\begin{aligned} \vdash R_1T & \frac{\vdash R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma}{\vdash R_{z,\bar{x},1}^{z,\bar{v},\bar{u}}(\Phi), \Sigma}; & \vdash R_1T & \frac{\vdash R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma}{\vdash R_{z,\bar{x},1}^{z,\bar{v},\bar{u}}(\Phi), \Sigma}; \\ \vdash \neg R_1T & \frac{\vdash \neg R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma}{\vdash \neg R_{z,\bar{x},1}^{z,\bar{v},\bar{u}}(\Phi), \Sigma}; & \vdash \neg R_1T & \frac{\vdash \neg R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma}{\vdash \neg R_{z,\bar{x},1}^{z,\bar{v},\bar{u}}(\Phi), \Sigma}; \\ \vdash \Phi_1N & \frac{\vdash R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma}{\vdash R_{z,\bar{x},1}^{y,\bar{v},\bar{u}}(\Phi), \Sigma}; & \vdash \Phi_1N & \frac{\vdash R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma}{\vdash R_{z,\bar{x},1}^{y,\bar{v},\bar{u}}(\Phi), \Sigma}; \\ \vdash \neg \Phi_1N & \frac{\vdash \neg R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma}{\vdash \neg R_{z,\bar{x},1}^{y,\bar{v},\bar{u}}(\Phi), \Sigma}; & \vdash \neg \Phi_1N & \frac{\vdash \neg R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma}{\vdash \neg R_{z,\bar{x},1}^{y,\bar{v},\bar{u}}(\Phi), \Sigma}. \end{aligned}$$

Для форм типу Φ_1N та $\neg \Phi_1N$ умова: $y \in v(\Phi)$.

$$\begin{aligned} \vdash R_1\exists p & \frac{\vdash \exists x\Phi, \Sigma}{\vdash R_y^x(\exists x\Phi), \Sigma}; & \vdash R_1\exists p & \frac{\vdash \exists x\Phi, \Sigma}{\vdash R_y^x(\exists x\Phi), \Sigma}; \\ \vdash \neg R_1\exists p & \frac{\vdash \neg \exists x\Phi, \Sigma}{\vdash \neg R_y^x(\exists x\Phi), \Sigma}; & \vdash \neg R_1\exists p & \frac{\vdash \neg \exists x\Phi, \Sigma}{\vdash \neg R_y^x(\exists x\Phi), \Sigma}; \\ \vdash R_1\exists R & \frac{\vdash R_{\bar{v},1}^{\bar{z},\bar{u}}(\exists xA), \Sigma}{\vdash R_{\bar{v},1,y}^{\bar{z},\bar{u},x}(\exists xA), \Sigma}; & \vdash R_1\exists R & \frac{\vdash R_{\bar{v},1}^{\bar{z},\bar{u}}(\exists xA), \Sigma}{\vdash R_{\bar{v},1,y}^{\bar{z},\bar{u},x}(\exists xA), \Sigma}; \\ \vdash \neg R_1\exists R & \frac{\vdash \neg R_{\bar{v},1}^{\bar{z},\bar{u}}(\exists x\Phi), \Sigma}{\vdash \neg R_{\bar{v},1,y}^{\bar{z},\bar{u},x}(\exists x\Phi), \Sigma}; & & \end{aligned}$$

$$\vdash \neg R_1\exists R \frac{\vdash \neg R_{\bar{v},1}^{\bar{z},\bar{u}}(\exists x\Phi), \Sigma}{\vdash \neg R_{\bar{v},1,y}^{\bar{z},\bar{u},x}(\exists x\Phi), \Sigma}.$$

Для форм типу $R_1\exists R, \neg R_1\exists R$ та $R_1\exists p, \neg R_1\exists p$ маємо: $y \in V \cup \{\perp\}$.

$$\begin{aligned} \vdash \neg \neg & \frac{\vdash \Phi, \Sigma}{\vdash \neg \neg \Phi, \Sigma}; & \vdash \neg \neg & \frac{\vdash \Phi, \Sigma}{\vdash \neg \neg \Phi, \Sigma}; \\ \vdash \vee & \frac{\vdash \Phi, \Sigma \quad \vdash \Psi, \Sigma}{\vdash \Phi \vee \Psi, \Sigma}; & \vdash \vee & \frac{\vdash \Phi, \Sigma \quad \vdash \Psi, \Sigma}{\vdash \Phi \vee \Psi, \Sigma}; \\ \vdash \neg \vee & \frac{\vdash \neg \Phi, \Sigma \quad \vdash \neg \Psi, \Sigma}{\vdash \neg(\Phi \vee \Psi), \Sigma}; & \vdash \neg \vee & \frac{\vdash \neg \Phi, \Sigma \quad \vdash \neg \Psi, \Sigma}{\vdash \neg(\Phi \vee \Psi), \Sigma}; \\ \vdash R_1R & \frac{\vdash R_{\bar{y},1}^{\bar{v},\bar{z}} \circ_{\bar{x},1}^{\bar{u},\bar{f}}(\Phi), \Sigma}{\vdash R_{\bar{y},1}^{\bar{v},\bar{z}}(R_{\bar{x},1}^{\bar{u},\bar{f}}(\Phi)), \Sigma}; & \vdash R_1R & \frac{\vdash R_{\bar{y},1}^{\bar{v},\bar{z}} \circ_{\bar{x},1}^{\bar{u},\bar{f}}(\Phi), \Sigma}{\vdash R_{\bar{y},1}^{\bar{v},\bar{z}}(R_{\bar{x},1}^{\bar{u},\bar{f}}(\Phi)), \Sigma}; \\ \vdash \neg R_1R & \frac{\vdash \neg R_{\bar{y},1}^{\bar{v},\bar{z}} \circ_{\bar{x},1}^{\bar{u},\bar{f}}(\Phi), \Sigma}{\vdash \neg R_{\bar{y},1}^{\bar{v},\bar{z}}(R_{\bar{x},1}^{\bar{u},\bar{f}}(\Phi)), \Sigma}; & \vdash \neg R_1R & \frac{\vdash \neg R_{\bar{y},1}^{\bar{v},\bar{z}} \circ_{\bar{x},1}^{\bar{u},\bar{f}}(\Phi), \Sigma}{\vdash \neg R_{\bar{y},1}^{\bar{v},\bar{z}}(R_{\bar{x},1}^{\bar{u},\bar{f}}(\Phi)), \Sigma}; \\ \vdash \neg R_1R & \frac{\vdash \neg R_{\bar{y},1}^{\bar{v},\bar{z}} \circ_{\bar{x},1}^{\bar{u},\bar{f}}(\Phi), \Sigma}{\vdash \neg R_{\bar{y},1}^{\bar{v},\bar{z}}(R_{\bar{x},1}^{\bar{u},\bar{f}}(\Phi)), \Sigma}; & \vdash \neg R_1R & \frac{\vdash \neg R_{\bar{y},1}^{\bar{v},\bar{z}} \circ_{\bar{x},1}^{\bar{u},\bar{f}}(\Phi), \Sigma}{\vdash \neg R_{\bar{y},1}^{\bar{v},\bar{z}}(R_{\bar{x},1}^{\bar{u},\bar{f}}(\Phi)), \Sigma}; \\ \vdash R_1\neg & \frac{\vdash \neg R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma}{\vdash R_{\bar{x},1}^{\bar{v},\bar{u}}(\neg \Phi), \Sigma}; & \vdash R_1\neg & \frac{\vdash \neg R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma}{\vdash R_{\bar{x},1}^{\bar{v},\bar{u}}(\neg \Phi), \Sigma}; \\ \vdash \neg R_1\neg & \frac{\vdash R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma}{\vdash \neg R_{\bar{x},1}^{\bar{v},\bar{u}}(\neg \Phi), \Sigma}; & \vdash \neg R_1\neg & \frac{\vdash R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma}{\vdash \neg R_{\bar{x},1}^{\bar{v},\bar{u}}(\neg \Phi), \Sigma}; \\ \vdash R_1\vee & \frac{\vdash R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma \quad \vdash R_{\bar{x},1}^{\bar{v},\bar{u}}(\Psi), \Sigma}{\vdash R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi \vee \Psi), \Sigma}; & \vdash R_1\vee & \frac{\vdash R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma \quad \vdash R_{\bar{x},1}^{\bar{v},\bar{u}}(\Psi), \Sigma}{\vdash R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi \vee \Psi), \Sigma}; \\ \vdash \neg R_1\vee & \frac{\vdash \neg R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma \quad \vdash \neg R_{\bar{x},1}^{\bar{v},\bar{u}}(\Psi), \Sigma}{\vdash \neg R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi \vee \Psi), \Sigma}; & \vdash \neg R_1\vee & \frac{\vdash \neg R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi), \Sigma \quad \vdash \neg R_{\bar{x},1}^{\bar{v},\bar{u}}(\Psi), \Sigma}{\vdash \neg R_{\bar{x},1}^{\bar{v},\bar{u}}(\Phi \vee \Psi), \Sigma}; \\ \vdash \exists \epsilon & \frac{\vdash R_z^x(\Phi), \vdash \epsilon z, \Sigma}{\vdash \exists x\Phi, \Sigma}; & \vdash \exists \epsilon & \frac{\vdash \neg R_z^x(\Phi), \vdash \epsilon z, \Sigma}{\vdash \neg \exists x\Phi, \Sigma}; \\ \vdash \exists R_1 & \frac{\vdash R_{\bar{v},1,z}^{\bar{u},\bar{w},x}(\Phi), \vdash \epsilon z, \Sigma}{\vdash R_{\bar{v},1}^{\bar{u},\bar{w}}(\exists x\Phi), \Sigma}; & \vdash \exists R_1 & \frac{\vdash R_{\bar{v},1,z}^{\bar{u},\bar{w},x}(\Phi), \vdash \epsilon z, \Sigma}{\vdash R_{\bar{v},1}^{\bar{u},\bar{w}}(\exists x\Phi), \Sigma}; \\ \vdash \neg \exists R_1\epsilon & \frac{\vdash \neg R_{\bar{v},1,z}^{\bar{u},\bar{w},x}(\Phi), \vdash \epsilon z, \Sigma}{\vdash \neg R_{\bar{v},1}^{\bar{u},\bar{w}}(\exists x\Phi), \Sigma}; & \vdash \neg \exists R_1\epsilon & \frac{\vdash \neg R_{\bar{v},1,z}^{\bar{u},\bar{w},x}(\Phi), \vdash \epsilon z, \Sigma}{\vdash \neg R_{\bar{v},1}^{\bar{u},\bar{w}}(\exists x\Phi), \Sigma}; \\ \vdash \exists f & \frac{\vdash \exists x\Phi, \vdash R_z^x(\Phi), \vdash \epsilon z, \Sigma}{\vdash \exists x\Phi, \Sigma}; & \vdash \exists f & \frac{\vdash \exists x\Phi, \vdash R_z^x(\Phi), \vdash \epsilon z, \Sigma}{\vdash \exists x\Phi, \Sigma}; \\ \vdash \neg \exists f & \frac{\vdash \neg \exists x\Phi, \vdash \neg R_z^x(\Phi), \vdash \epsilon z, \Sigma}{\vdash \neg \exists x\Phi, \Sigma}; & \vdash \neg \exists f & \frac{\vdash \neg \exists x\Phi, \vdash \neg R_z^x(\Phi), \vdash \epsilon z, \Sigma}{\vdash \neg \exists x\Phi, \Sigma}; \end{aligned}$$

$$\vdash_{\perp} \exists R_{\perp} f \frac{\vdash_{\perp} R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \vdash_{\perp} R_{\perp, \perp, z}^{\bar{u}, \bar{w}, x} (\Phi), \vdash_{\perp} \varepsilon z, \Sigma}{\vdash_{\perp} R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Sigma};$$

$$\vdash_{\neg} \exists R_{\perp} f \frac{\vdash_{\neg} R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \vdash_{\neg} R_{\perp, \perp, z}^{\bar{u}, \bar{w}, x} (\Phi), \vdash_{\neg} \varepsilon z, \Sigma}{\vdash_{\neg} R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Sigma}.$$

Для форм $\vdash \exists \varepsilon$, $\vdash_{\neg} \exists \varepsilon$, $\vdash f$, $\vdash_{\neg} \exists f$ умова: $z \in V_T$, $z \notin nm(\Sigma, \exists x \Phi)$.

Для форм $\vdash \exists R_{\perp} \varepsilon$, $\vdash_{\neg} \exists R_{\perp} \varepsilon$, $\vdash R_{\perp} f$, $\vdash_{\neg} \exists R_{\perp} f$ умова: $z \in V_T$, $z \notin nm(\Sigma, R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi))$.

$\vdash f$, $\vdash_{\neg} \exists f$, $\vdash R_{\perp} f$, $\vdash_{\neg} \exists R_{\perp} f$ – це форми типу $\exists f$. Для цих форм Σ не містить формул вигляду $\vdash \varepsilon z$.

$$\vdash_{\exists} \frac{\vdash_{\exists} \exists x \Phi, \vdash_{\exists} R_y^x (\Phi), \vdash_{\exists} \varepsilon y, \Sigma}{\vdash_{\exists} \exists x \Phi, \vdash_{\exists} \varepsilon y, \Sigma};$$

$$\vdash_{\neg} \exists \frac{\vdash_{\neg} \exists x \Phi, \vdash_{\neg} R_y^x (\Phi), \vdash_{\neg} \varepsilon y, \Sigma}{\vdash_{\neg} \exists x \Phi, \vdash_{\neg} \varepsilon y, \Sigma};$$

$$\vdash_{\perp} \exists R_{\perp} \frac{\vdash_{\perp} R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \vdash_{\perp} R_{\perp, \perp, y}^{\bar{u}, \bar{w}, x} (\Phi), \vdash_{\perp} \varepsilon y, \Sigma}{\vdash_{\perp} R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \vdash_{\perp} \varepsilon y, \Sigma};$$

$$\vdash_{\neg} \exists R_{\perp} \frac{\vdash_{\neg} R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \vdash_{\neg} R_{\perp, \perp, y}^{\bar{u}, \bar{w}, x} (\Phi), \vdash_{\neg} \varepsilon y, \Sigma}{\vdash_{\neg} R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \vdash_{\neg} \varepsilon y, \Sigma}.$$

$\vdash_{\exists} \varepsilon$, $\vdash_{\neg} \exists \varepsilon$, $\vdash R_{\perp} \varepsilon$, $\vdash_{\neg} \exists R_{\perp} \varepsilon$ – це форми типу $\exists \varepsilon$.

$$\vdash_{\exists} d \frac{\vdash_{\exists} \varepsilon y, \vdash_{\exists} \exists x \Phi, \Sigma \quad \vdash_{\exists} \exists x \Phi, \vdash_{\exists} R_y^x (\Phi), \vdash_{\exists} \varepsilon y, \Sigma}{\vdash_{\exists} \exists x \Phi, \Sigma};$$

$$\vdash_{\neg} \exists d \frac{\vdash_{\neg} \varepsilon y, \vdash_{\neg} \exists x \Phi, \Sigma \quad \vdash_{\neg} \exists x \Phi, \vdash_{\neg} R_y^x (\Phi), \vdash_{\neg} \varepsilon y, \Sigma}{\vdash_{\neg} \exists x \Phi, \Sigma}.$$

Форми $\vdash_{\perp} \exists R_{\perp} d$ та $\vdash_{\neg} \exists R_{\perp} d$ мають такий вигляд:

$$\vdash_{\perp} \varepsilon y, \vdash_{\perp} R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Sigma \quad \vdash_{\perp} R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \vdash_{\perp} R_{\perp, \perp, y}^{\bar{u}, \bar{w}, x} (\Phi), \vdash_{\perp} \varepsilon y, \Sigma \quad \vdash_{\perp} R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Sigma;$$

$$\vdash_{\neg} \varepsilon y, \vdash_{\neg} R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Sigma \quad \vdash_{\neg} R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \vdash_{\neg} R_{\perp, \perp, y}^{\bar{u}, \bar{w}, x} (\Phi), \vdash_{\neg} \varepsilon y, \Sigma \quad \vdash_{\neg} R_{\perp, \perp}^{\bar{u}, \bar{w}} (\exists x \Phi), \Sigma.$$

$\vdash_{\exists} d$, $\vdash_{\neg} \exists d$, $\vdash R_{\perp} d$, $\vdash_{\neg} \exists R_{\perp} d$ – це форми типу $\exists d$.

Для цих форм умова: εy не входить до Σ та Σ містить символи вигляду εz .

$\vdash \exists R_{\perp} \varepsilon$, $\vdash_{\neg} \exists R_{\perp} \varepsilon$, $\vdash \varepsilon$, $\vdash_{\neg} \varepsilon$ назвемо $\exists_T$ -формами, форми типів $\exists f$, $\exists \varepsilon$, $\exists d$ назвемо $\exists_F$ -формами.

$Q_{\perp}L$ -числення формалізує відношення $\models_T$ для випадку неокласичної семантики.

Умова замкненості секвенції: $C \vee CL \vee UnC$.

Базові секвенційні форми такі ж, як наведені вище форми числення $Q_{\perp}G$.

$Q_{\perp}R$ -числення формалізує відношення $\models_F$ для випадку неокласичної семантики. (

Умова замкненості секвенції: $C \vee CR \vee UnC$.

Базові секвенційні форми такі ж, як наведені вище форми числення $Q_{\perp}G$.

$Q_{\perp}LR$ -числення формалізує відношення $\models_{TF}$ для випадку неокласичної семантики.

Умова замкненості секвенції: $C \vee CLR \vee UnC$.

Базові секвенційні форми такі ж, як наведені вище форми числення $Q_{\perp}G$.

$Q_{\perp}C$ -числення формалізує відношення $\models_{CI}$ для випадку неокласичної семантики.

Умова замкненості секвенції: $C \vee UnC$.

Базові секвенційні форми $Q_{\perp}C$ -числень аналогічні відповідним формам (див. [5]) QSC -числень:

$$\vdash R_{\perp} T, \vdash R_{\perp} \neg, \vdash \Phi_{\perp} N, \vdash \neg \Phi_{\perp} N, \vdash R_{\perp} \exists p, \vdash R_{\perp} \exists p, \vdash R_{\perp} \exists R, \vdash R_{\perp} \exists R, \vdash \neg, \vdash \neg, \vdash \vee, \vdash \vee, \vdash R_{\perp} R, \vdash R_{\perp} R, \vdash R_{\perp} \neg, \vdash R_{\perp} \neg, \vdash R_{\perp} \vee, \vdash R_{\perp} \vee, \vdash \varepsilon, \vdash \exists R_{\perp} \varepsilon, \vdash f, \vdash \exists R_{\perp} f, \vdash \exists v, \vdash \exists R_{\perp} v, \vdash d, \vdash \exists R_{\perp} d.$$

Зазначені форми, окрім $\vdash \neg$ та $\vdash \neg$, наведені вище. Форми $\vdash \neg$ та $\vdash \neg$ мають традиційний вигляд:

$$\vdash \neg \frac{\vdash \Phi, \Sigma}{\vdash \neg \Phi, \Sigma}; \quad \vdash \neg \frac{\vdash \Phi, \Sigma}{\vdash \neg \Phi, \Sigma}.$$

Основна властивість секвенційних форм збудованих числень ЧКНЛРР впливає з властивостей відношень $\models$, вона така ж, як для числень ЧКНЛ:

$$\text{Теорема 2. Нехай } \frac{\vdash \Lambda_{\perp} K}{\vdash \neg \Gamma \Delta} \text{ та } \frac{\vdash \Lambda_{\perp} K \quad \vdash X_{\perp} Z}{\vdash \neg \Gamma \Delta} -$$

базові секвенційні форми. Тоді маємо:

- 1) $\Lambda \models K \Rightarrow \Gamma \models \Delta$; $\Lambda \models K$ та $X \models Z \Rightarrow \Gamma \models \Delta$;
- 2) $\Gamma \not\models \Delta \Rightarrow \Lambda \not\models K$; $\Gamma \not\models \Delta \Rightarrow \Lambda \not\models K$ або $X \not\models Z$.

Таким чином, для ЧКНЛРР часткових предикатів побудовано наступні секвенційні числення.

Таблиця 1. Секвенційні числення ЧКНЛРР.

	$\models_{CI}$	$\models_T$	$\models_F$	$\models_{TF}$
Неокласична семантика	$Q_{\perp}C$	$Q_{\perp}L$	$Q_{\perp}R$	$Q_{\perp}LR$
Загальна семантика	–	$Q_{\perp}G$	$Q_{\perp}G$	$Q_{\perp}G$

Процедура побудови секвенційного дерева для збудованих числень в цілому така ж, як процедура побудови дерева для відповідних числень ЧКНЛ (див., напр., [1, 6]). Якщо процедура побудови дерева завершена позитивно, то маємо скінченне замкнене дерево, всі його вершини – замкнені секвенції. Якщо процедура завершена негативно (тоді маємо скінченне незамкнене дерево) або процедура не завершується (будоване дерево нескінченне), то у дереві існує незамкнений шлях, всі його вершини – незамкнені секвенції..

Теорема 3 (коректності). Секвенція $\vdash \neg \Gamma \Delta$ відна в численні $\beta \Rightarrow \Gamma \models \Delta$ в семантиці σ .

Назву β числення читаємо на перетині стовпця $\models$ та рядка σ (табл. 1).

Справді, якщо $\vdash \Gamma \vdash \Delta$ вивідна, тоді для неї побудоване замкнене секвенційне дерево. Для кожної його вершини $\vdash \Lambda \vdash K$ маємо $\Lambda \models K$. Для листів дерева це впливає з їх замкненості. Збереження секвенційними формами відношення $\models$ (від засновків до висновків) впливає із теореми 1.

Повнота побудованих числень впливає із відповідних теорем про існування контрмоделі для множини формул незамкненого шляху секвенційного дерева. Доведення цих теорем опирається на метод модельних множин (див. [1, 5, 6]).

Теорема 4 (про контрмоделі для числень Q_1G, Q_1L, Q_1R, Q_1LR). Нехай $\wp$ – незамкнений шлях у секвенційному дереві, збудованому для $\vdash \Gamma \vdash \Delta$, нехай H – множина всіх специфікованих формул секвенцій цього шляху. Тоді існують моделі мови $A = (A, I), B = (A, I)$ та $\delta, \eta \in {}^VA$:

- 1) $\vdash \Phi \in H \Rightarrow \delta \in T(\Phi_A)$ та $\vdash \Phi \in H \Rightarrow \delta \notin T(\Phi_A)$;
- 2) $\vdash \Phi \in H \Rightarrow \eta \in F(\Phi_B)$ та $\vdash \Phi \in H \Rightarrow \eta \notin F(\Phi_B)$.

Такі пари (A, δ) та (B, η) назовемо T -контрмоделлю та F -контрмоделлю для секвенції $\vdash \Gamma \vdash \Delta$.

Теорема 5 (про контрмоделі для числення Q_1C). Нехай $\wp$ – незамкнений шлях у секвенційному дереві для $\vdash \Gamma \vdash \Delta$, нехай H – множина всіх специфікованих формул секвенцій цього шляху. Тоді існує модель мови $A = (A, I)$ та $\delta \in {}^VA$:

$$\vdash \Phi \in H \Rightarrow \delta \in T(\Phi_A) \text{ та } \vdash \Phi \in H \Rightarrow \delta \in F(\Phi_A);$$

Така (A, δ) – CI -контрмодель для секвенції $\vdash \Gamma \vdash \Delta$.

Для різних числень та логічних наслідків теорема повноти формулюється однотипно. Вона доводиться традиційним чином (див., напр., [5, 6]) на базі відповідних теорем про контрмоделі.

Теорема 6 (повноти). $\Gamma \models \Delta$ в семантиці $\sigma \Rightarrow$ секвенція $\vdash \Gamma \vdash \Delta$ вивідна в численні β .

Поєднаємо теореми коректності та повноти:

Теорема 7. $\Gamma \models \Delta$ в семантиці $\sigma \Leftrightarrow$ секвенція $\vdash \Gamma \vdash \Delta$ вивідна в численні β .

Назву β числення читаємо на перетині стовпця $\models$ та рядка σ (табл. 1).

Висновки

В роботі досліджено нові класи першопорядкових композиційно-номінативних логік часткових однозначних предикатів та часткових неод-

нозначних предикатів. Особливістю цих логік є використання композицій розширеної реномінації, що дає змогу явно задавати відсутність значення для предметних імен. Розглянуто властивості відношень логічного наслідку для множин формул таких логік. Для формалізації таких відношень запропоновано числення секвенційного типу. Доведено теореми коректності та повноти побудованих числень.

Дослідження першопорядкових композиційно-номінативних логік з розширеними реномінаціями планується продовжити в наступних роботах.

Список використаних джерел

1. Nikitchenko M.S., Shkilniak S.S. Mathematical Logic and Theory of Algorithms. – Kyiv: VPC Kyivskyi Universytet, 2008. – 528 p. (in Ukrainian).
2. Nikitchenko M.S., Shkilniak S.S. First-Order Composition-Nominative Logics // Visnyk, Ser. Fiz.-Mat. Nauky, Kyiv Univ. im. Tarasa Shevchenka. – 2011. – № 4. – P. 176–185 (in Ukrainian).
3. Nikitchenko M.S., Shkilniak O.S., Shkilniak S.S. Logics of partial predicates with extended renominations and quantifier // Visnyk, Ser. Fiz.-Mat. Nauky, Kyiv Univ. im. Tarasa Shevchenka. – 2013. – № 2. – P. 210–215 (in Ukrainian).
4. Shkilniak S.S. Semantic properties of partial predicates with extended renominations and quantifier // Visnyk, Ser. Fiz.-Mat. Nauky, Kyiv Univ. im. Tarasa Shevchenka. – 2013. – № 3. – P. 297–302 (in Ukrainian).
5. Shkilniak S.S. Sequent calculi for logics of partial and multi-valued quasi-ary predicates // Visnyk, Ser. Fiz.-Mat. Nauky, Kyiv Univ. im. Tarasa Shevchenka. – 2012. – № 4. – P. 231–236 (in Ukrainian).
6. Shkilniak S.S. Sequent Calculi of Composition Nominative Logics of Quasi-ary Predicates // Problems in Programming. – 2012. – № 2–3. – P. 33–43 (in Ukrainian).

Надійшла до редколегії 10.09.2013

Підписано до друку _____. Формат _____. Друк. офс.
Папір офс. № _____. Друк. арк. _____. Тираж _____. Зам. _____.
Друкарня _____.