Technology Designer: Master Manual // Research Institute of Information Resources at National Technical University of Ukraine "Kiev Politechnic Institute", Kiev, Ukraine; MicroSystem Lab. at Sunmoon University, Asan, Chungnan, Korea. – 1998. – 453 p.

УДК 004.4'232

Я. Драґан[1], В. Овсяк[2], О. Овсяк[3]
[1]Національний університет Львівська політехніка,
кафедра програмного забезпечення,
[2]Українська академія друкарства,
[3]Київський національний університет культури і мистецтв

ПОРІВНЯННЯ АЛГЕБРИЧНИХ МЕТОДІВ ПОДАНЬ АЛГОРИТМІВ

Порівняннями за системами операцій та їхніми властивостями та побудованими моделями абстрактної комп'ютерної системи дано оцінку таких відомих алгебричних методів опису алгоритмів, якими є модифікована система алгоритмічних алгебр, модифікована алгебра алгоритмів та алгебра алгоритмів.

Ключові слова: операція, властивість, алгебра, система, декомпозиція, модель.

Bi comparison under the systems operations and their properties and constructed abstract models of computer systems estimations of such well-known algebraic methods for describing algorithms, which are modified systems of algorithmic algebras, modified algebra algorithms and algebra algorithms are given.

Key words: operation, property, algebra, system, decomposition, model.

Вступ

Сучасні комп'ютерні системи є складними системами. Процес їхнього проектування передбачає створення математичного і програмного забезпечення. Для розроблення як математичного, так і програмного забезпечення застосовується метод розбиття. Як правило, розбиття є багаторівневим і виконується за вибраними критеріями. Одним із таких важливих критеріїв є функціональне призначення підсистем.

Синтез математичного забезпечення комп'ютерних систем загалом, а також їхніх складових, на які вони розбиті, можна виконати із застосуванням модифікованої системи алгоритмічних алгебр [1, 2] і модифікованої алгебри алгоритмів [3–5]. Застосування алгебричних методів для опису математичного забезпечення систем, крім точного опису моделей, створює передумови для їхньої оптимізації та дослідження вірогідності.

Порівняння алгебри алгоритмів з модифікованою алгеброю алгоритмів і модифікованої системи алгоритмічних алгебр з модифікованою алгеброю алгоритмів сприятиме вибору цих засобів для розв'язання задач синтезу, оптимізації за вибраними критеріями і дослідження математичного забезпечення комп'ютерних систем.

Характеристика здобутків і формулювання завдання дослідження

Наш час – початок XXI століття, як і кінець попереднього, дослідники звикли називати ерою інформатики і комп'ютерно-інформатичних технологій, маючи на увазі вдосконалення та повсюдне використання широкого арсеналу засобів допомоги людині не стільки у виконанні фізичної праці, як, головно, розумової діяльності, тобто дій, властивих людському мисленню. Універсальні обчислювальні машини досконало пристосовані до розв'язання цілком означених обчислювальних задач. Труднощі полягають у тому, як використовувати доступні чи розробляти нові потрібні алгоритми розв'язання їх, коли під алгоритмом розуміють множину цілком означених правил розв'язання за скінченну кількість кроків [6].

З давніх часів багато задач математики полягало в пошуку тих чи інших конструктивних способів їх розв'язання. Це вже вироблення зручної символіки й усвідомлення факту відсутності потрібних методів привело до створення конструктивного напряму в математиці, хоча і до цього дослідникам траплялись задачі, які потребували вміння розв'язати їх "у загальному випадку", а це означало, по суті, володіння певним придатним для цього алгоритмом – нехай і вираженим описово (тобто звичайною мовою).

І починаючи зі знаменитої тепер праці "Про числа і дії над ними" Мухамада бен Муси аль Хорезмі, перекладеної з арабської мови латинською у XII ст., яка познайомила європейців з індійсько-арабським десятковим позиційним способом числення, з'явився відомий нам термін "алгоритм". Це змінена під впливом грецького *аритмос* (αριτμος число) форма латинізованого імені аль Хорезмі (із Хорезму). Спочатку цим словом називали правила чотирьох дій арифметики (у десятковій системі числення). І це було цілком слушно. Але згодом про це начисто забули і починають історію з так званого алгоритму Евкліда (див. відп. статті у словнику [6] і довідниках [7, 8]). Інша праця аль Хорезмі започаткувала алгебру як окрему гілку математики, а слово походить з латинізованої назви "аль-джебр" операції, яку він використовував, розв'язуючи рівняння.

Поява неевклідової геометрії стимулювала зацікавлення дедуктивною побудовою розділів математики, а Д. Гільберт навіть запропонував програму так званого аксіоматичного обґрунтування всієї математики, спершу зводячи несуперечність геометрії до несуперечності арифметики. Але результати К. Ґеделя (30-ті рр. XX ст.) привели до краху цієї надії (про апріоризм аксіоматичного методу і конечність залучення інфінітних методів (див. [9], де узагальнено публікації [10, 11]).

Водночас при аналізі поняття "алгоритм" і математичних моделей біологічних систем з'явились машини Тюрінґа, Поста та наслідувачів їх – аж до Маркова й Колмогорова, а також так звані абстрактні автомати. Перші з них є перетворювачами даних дискретної дії, а другі – фактично алгебричними системами з вхідною та вихідною абетками, сукупністю станів і двома функціями переходів (поз. [11]), які теж запроваджено з метою навести лад у цій проблемі. Але оскільки поняття алгоритму належить до основних і не допускає означення через простіші чи загальніші, то такі машини трактували (запроваджуючи їх) як певні "уточнення" поняття алгоритму. Підкреслимо принагідно, що така назва обманлива, бо ніхто ніяких таких машин ніколи реально не будував (хіба що Ляйбніц 1671 р. побудував першу відому машину множення [12]), а це були умоглядні схеми для розв'язання внутрішньоматематичних проблем. І як підкреслює В.Успенський у статті "Алгоритм" (див. [6]), ці машини загалом мети не досягли, бо гіпотеза (Тюрінґа – Черча), що для кожного типу уточнення поняття алгоритму можна вказати відповідний алгоритм, "на сучасному рівні наших знань не може бути предметом математичного доведення", і всі запроваджені у такий спосіб уточнення "прийнято вважати у певному природному сенсі еквівалентними" (доведено [13] тільки рівносильність рекурсивних функцій та розроблених 1947 р. спеціально так званих нормальних алгорифмів (!)). А уточненням вважають поняття масової алгоритмічної проблеми, коли шукають єдиний алгоритм розв'язання окремих задач певного типу. Якщо його не існує, то кажуть, що проблема алгоритмічно нерозв'язна.

Напрям досліджень, стимульований "машинами", крім внеску у проблему основ математики, чітко виявив поняття алгоритмічної нерозв'язності (як, наприклад, десятої проблеми Гільберта з теорії діофантових рівнянь), що істотно змінило уявлення про алгоритмічну проблему.

Свого роду крапку в "машинній проблемі" ставить думка А.Колмогорова (одного з авторів "уточнення" поняття алгоритму) із статті "Математика" – підсумкової у словнику [6] (с. 7 – 37): "ніяка єдина дедуктивна теорія не може вичерпати проблеми теорії чисел … навіть при розвої арифметики натуральних чисел … Цьому в теорії алгоритмів відповідають теореми про неможливість універсальних алгоритмів для досить широких класів математичних проблем. Ці теореми дали філософії математики найцікавішу гостру конкретизацію положення про те, що живе мислення принципово різниться від роботи всякого виду обчислювальних автоматів”. Таку ж тезу він висловлював ще у доповіді “Автомати й життя” (докладніше див. [10]). Аналогічно Ґ. Біркгоф [12] наголошує, що цифрові машини не моделюють людську уяву, а вона “конечна (необходимая) для формулювання найглибших ідей навіть у чистій математиці, особливо в аналізі, і для всякої значної роботи в математиці прикладній”. Зокрема, показовою для цього є анкета Ж. Адамара й аналіз її (див. [14]). Адамар, покликаючись на А. Пуанкаре, підкреслює й ілюструє прикладами роль інтуїції, свідомості та підсвідомості у математичній творчості. Біркгоф виділяє думку Уайтгеда й Расела, що “всяка теорія принципів математики має бути індуктивною”, та ілюструє її тим, що тезу Тюрінґа неможливо ні довести, ні заперечити, оскільки вона твердить, що всяке “означене” дійсне число може бути побудоване якоюсь машиною Тюрінґа, а слово “означене” само неозначне. Далі Біркгоф наголошує, що здібність доброго математика відчувати суттєве погано піддається механізації, а без неї машині доведеться робити багато зайвого, чого уникає досвідчений математичний розум. Тому конечним є симбіоз людини й машини. Це перегукується з концепцією людино-машинних комплексів В.Глушкова.

Але під час аналізу реальних обчислювальних машин зроблено крен у так звану алгоритміку яко галузь перетворень алгоритмів шляхом дій над ними, коли об’єктом уваги слугують високорівневі моделі алгоритмів і програм [1, 2], чим залишено поза увагою питання базових алгоритмів (конструювання їх та перетворення) яко поля дій алгебри алгоритмів (див. [9]).

Вона виникла як вислід синтезу ідеї Ґ.Фреґе виходу поза лінійність та досвіду фізиків і привела до відкриття нової можливості шляхом формалізації збережності порядку застосування окремих операторів – компонент алгоритмів при алгебричних перетвореннях композитних алгоритмів з використанням розробленої системи ранжування (індексації) та процедур перетворення індексів [15].

Поява цієї алгебри підтверджує тезу А.Єршова (див. [6]): “Загалом побудова алгоритмів є творчим процесом, що вимагає глибоких знань предметної області і здатності до здогаду”.

Із загального погляду під час розроблення способів і засобів розв’язання потрібних задач мають справу з так званим об’єктно-орієнтованим та проблемно-орієнтованим програмуванням. А це переводить в область побудови чи радше обґрунтування математичних моделей – математичних об’єктів як мислених конструкцій, що втілюють у своїй структурі істотні для розв’язуваних задач властивості досліджуваних об’єктів. Це значить, що об’єкт і задача “диктують” (тому, хто здатний це вловити) формалізацію – математичну модель об’єкта для задачі. Модель має бути математичною, щоб уможливити використання засобів математики для розв’язання задачі. А тоді вже структура моделі “диктує” алгоритм, який вже в очевидний спосіб “диктує” програмну реалізацію.

Тому в фізико-технічних науках такою суттєвою є концепція МАПР – тріяди: модель – алгоритм – програмна реалізація з визначальною роллю в ній саме моделі. Вона є підставою доведення, що побудований на ній алгоритм розв’язує задачу, та адаптації алгоритму як загального до конкретнішої задачі як результату дослідження математичної структури моделі. Підкреслімо, що конкретизацію й адаптацію забезпечує системний аналіз усієї проблеми, який засновує своє трактування досліджуваного об’єкта на пріоритеті задачі на противагу прикладній математиці з пріоритетом методів у її основі, а комп’ютика – навіть надає пріоритет засобам у такій загостреній формі, що їй надав Дж. Клір з його автоматичним розв’язувачем системних задач (див. [9]).

Наведені тут підсумки системного аналізу розвою концепції алгоритмів показують природність алгебричних методів дослідження їх. А це зумовлює актуальність порівняння й оцінення можливостей цих засобів, спираючись на загальну концепцію праксеології Слуцького – Котарбінського (див. [11]), зокрема здобутків напряму алгебри алгоритмів.

Системи операцій та базові властивості операцій алгебри алгоритмів і модифікованої алгебри алгоритмів

Можливості алгебри визначаються базовою множиною системи операцій та їхніми властивостями. Тому наведемо порівняння алгебр за цими критеріями.

У табл. 1 подано операції та базові властивості операцій алгебри алгоритмів і модифікованої алгебри алгоритмів за системами операцій та їхніми властивостями. Знак “+” означає наявність операції або властивості, а знак “–” – їхню відсутність.

Таблиця 1

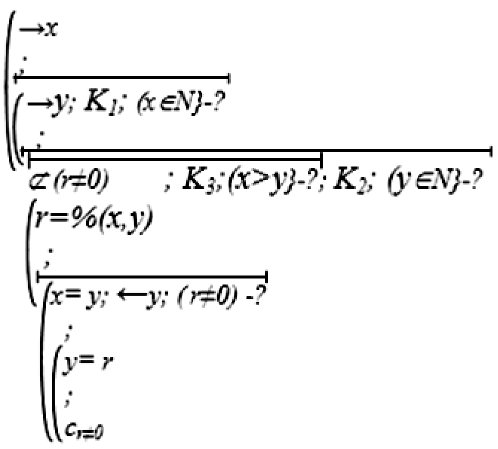
Системи операцій та базові властивості операцій алгебри алгоритмів і модифікованої алгебри алгоритмів

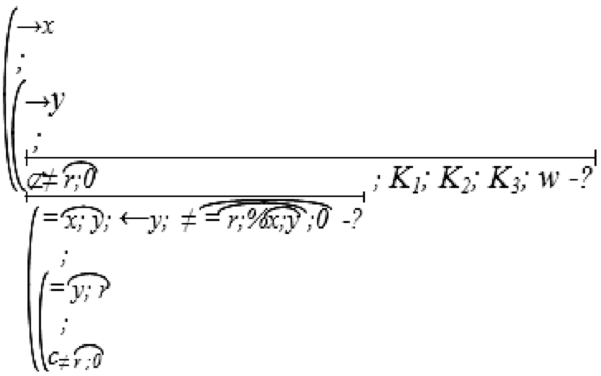
№	Операція, властивість	Алгебра алгоритмів	Модифікована алгебра алгоритмів
1	*Секвентування*	+	+
2	*Елімінування*	+	+
3	*Паралелення*	+	+
4	*Реверсування*	+	+
5	*Циклічне секвентування*	+	+
6	*Циклічне елімінування*	+	+
7	*Циклічне паралелення*	+	+
8	*Багатозначне елімінування*	–	+
9	*Зв'язок між секвентуванням та елімінуванням*	–	+
10	*Зв'язок між секвентуванням, паралеленням та елімінуванням*	–	+
11	*Зв'язок між циклічним секвентуванням і паралеленням*	–	+
12	*Зв'язок між циклічним паралеленням і секвентуванням*	–	+
13	*Зв'язок між циклічним елімінуванням та його реверсуванням*	–	+
14	*Операції над логічними сталими*	–	+

Модифікована алгебра алгоритмів, як і алгебра алгоритмів, містить операції секвентування, елімінування, паралелення, реверсування, циклічного секвентування, циклічного елімінування і циклічного паралелення.

В алгебрі алгоритмів нема операції багатозначного елімінування, яка є у модифікованій алгебрі алгоритмів. Крім того, в алгебрі алгоритмів відсутні властивості встановлення взаємозв’язків між операціями секвентування та елімінування, секвентуванням, паралеленням та елімінуванням, циклічним секвентуванням і паралеленням, циклічним паралеленням і секвентуванням, циклічним елімінуванням і його реверсуванням, а також операції над логічними сталими, які є у модифікованій алгебрі алгоритмів.

Кожна із наведених нижче формул (ліва і права) описує знаходження найбільшого спільного дільника двох натуральних чисел:





де →*x* – символ введення першого числа і приписування його змінній *x*; аналогічно описуємо введення і приписування другого числа →*y*; перевірку належності (∈) значень змінних *x* і *y* до натуральних чисел *N* запишемо як (*x*∈*N*)*-?* і (*y*∈*N*)*-?*; K_1 і K_2 – повідомлення на екран комп’ютера про те, що змінні не належать до натуральних чисел; K_3 – повідомлення про те, що натуральне число *x* не є більшим за натуральне число *y*; *%(x, y)* – знаходження остачі від ділення, а *r=%(x, y)* – приписування остачі змінній *r*; ← *y* – вивід на екран комп’ютера значення найбільшого спільного дільника.

Права формула, яка має багатозначне елімінування, містить вдвічі менше операцій елімінування.

Тепер порівняймо модифіковану систему алгоритмічних алгебр і модифіковану алгебру алгоритмів за їхніми системами операцій. Операції модифікованої системи алгоритмічних алгебр і модифікованої алгебри алгоритмів наведено у табл. 2. Модифікована алгебра алгоритмів має такі знаки операцій: ―– реверсування, ⌒– секвентування, ⊢⊣ – елімінування, ⌐¬– паралелення, ⊄ – циклічного секвентування, ⊅ – циклічного елімінування і Ø – циклічного паралелення.

Таблиця 2

Операції модифікованої системи алгоритмічних алгебр і модифікованої алгебри алгоритмів

№	Модифікована система алгоритмічних алгебр		Модифікована алгебра алгоритмів	
	Назва операції	Позна-ня	Назва операції	Позна-ня
1	*кон’юнкція*	*x&y*	–	
2	*диз’юнкція*	*X|Y*	–	
3	*не*	$\overline{x}$	*реверсування*	$\overline{\ }\ x$
4	*прогнозування*	*X•u*	–	
5	*композиція*	*X*Y*	*секвентування*	$\overset{\frown}{X;Y}$
6	*альтернатива*	*([u] X, Y)*	*елімінування*	*X;Y;u*
7	*цикл*	*{[u] X}*	*циклічне секвентування*	*⊄uX*
8	–		*циклічне елімінування*	*⊅ uX*
9	–		*циклічне паралелення*	*ØuX*
10	*фільтр*	*F(u)*	–	
11	*асинхронна диз'юнкція*	*X||Y*	*паралелення*	*X;Y*
12	*контрольна точка*	*T(u)*	–	
13	*синхронізатор*	*S(u)*	–	

Модифікована система алгоритмічних алгебр є двоосновною, утвореною логічною і операторною основами. Вона містить логічні операції кон'юнкції, диз'юнкції, інвертування (не) і прогнозування та операторні операції – композиція, альтернатива, цикл, фільтр, асинхронна диз'юнкція, контрольна точка, синхронізатор.

У модифікованій алгебрі алгоритмів є операції секвентування, елімінування, паралелення, реверсування, циклічного секвентування, циклічного елімінування і циклічного паралелення. Вона не має операцій кон'юнкції і диз'юнкції, прогнозування, фільтра, контрольної точки і синхронізатора, які є у модифікованій системі алгоритмічних алгебр. Натомість у модифікованій алгебрі алгоритмів є операції багатозначного елімінування, циклічного паралелення і циклічного елімінування, зв'язки між операціями (див. табл. 3).

Таблиця 3

Порівняння методів алгебричного подання алгоритмів за показниками

Показники	Модифікована система алгоритмічних алгебр	Модифікована алгебра алгоритмів
Операція циклічного паралелення	–	+
Операція циклічного елімінування	–	+
Багатозначне елімінування	–	+
Зв'язки між операціями	–	+
Операції алгебри Буля	+	–
Асоціативність операцій	+	–/+
Поліморфізм алгоритмів	+	+
Опис асоціативних алгоритмів	+	+
Опис неасоціативних алгоритмів	–	+
Наявність індексів порядку	–	+

Для подання послідовностей у модифікованій системі алгоритмічних алгебр застосовують операцію композиції. Однак операція композиції є асоціативною [1]. У такому разі неасоціативні алгоритми неможливо описати системою алгоритмічних алгебр та модифікованою системою алгоритмічних алгебр.

В алгебрі алгоритмів і модифікованій алгебрі алгоритмів для подання послідовностей застосовується операція секвентування, яка, у загальному випадку, є неасоціативною. У зв'язку з цим алгеброю алгоритмів і модифікованою алгеброю алгоритмів формалізуються як асоціативні, так і неасоціативні алгоритми.

Розгляньмо застосування модифікованої алгебри алгоритмів і модифікованої системи алгоритмічних алгебр для побудови моделі абстрактної комп'ютерної системи з інтерфейсом користувача.

Модель абстрактної комп'ютерної системи

З метою зменшення складності побудови моделі абстрактної комп'ютерної системи застосовуємо метод декомпозиції. Декомпозиція виконується за певними критеріями, якими, наприклад, можуть бути функціональне призначення, вимоги до системи та її складових, їхні особливості тощо.

Підсистемна декомпозиція призначена для отримання складових системи, якими у випадках систем з інтерфейсами користувачів є підсистеми *функціонального* і *графічно-функціонального* призначення.

Візуальноелементні та *елементновластивістні* складові є результатом декомпозиції підсистем графічно-функціонального призначення.

Візуальноелементна декомпозиція утворена, як правило, елементами інтерфейсу різного призначення і є багаторівневою.

Елементновластивістна декомпозиція є текстовим описом візуальноелементної декомпозиції.

Функціональне поєднання візуальноелементних та елементновластивістних складових є у функціональних підсистемах, декомпозиція яких утворена змінними, структурами, властивостями, процедурами тощо.

Процедури утворено змінними, відношеннями, операціями, властивостями.

1. *Чотирирівнева підсистемна декомпозиція* абстрактної комп’ютерної системи, описана модифікованою алгеброю алгоритмів, матиме такий вигляд:

$$S = \varnothing i\, \overline{P_i = \varnothing j_i \overline{Q_{j,i} = \varnothing t_{j,i} \overline{F_{t,j,i} = \varnothing x_{t,j,i} H}}}_{x,t,j,i},$$

де $i \in \{0, 1, ...\ p\text{-}1\}$; $j \in \{0, 1, ...\ q\text{-}1\}$; $t \in \{0, 1, ...\ s\text{-}1\}$; $x \in \{0, 1, ...\ h\text{-}1\}$. Ця формула має 41 знак.

2. Ця сама чотирирівнева декомпозиція, виражена модифікованою системою алгоритмічних алгебр, є такою:

$$S = (P_0 = (Q_{0,0} = (F_{0,0,0} = (H_{0,0,0,0} \| \dots H_{h\text{-}1,0,0,0}) \| \dots F_{s\text{-}1,0,0} = (H_{0,s\text{-}1,0,0} \| \dots H_{h\text{-}1,s\text{-}1,0,0})) \| \dots$$
$$Q_{q\text{-}1,0} = (F_{0,\,q\text{-}1,0} = (H_{0,0,\,q\text{-}1,0} \| \dots H_{h\text{-}1,0,\,q\text{-}1,0}) \| \dots F_{s\text{-}1,\,q\text{-}1,0} = (H_{0,s\text{-}1,\,q\text{-}1,0} \| \dots H_{h\text{-}1,s\text{-}1,\,q\text{-}1,0}))) \| \dots$$
$$(P_{p\text{-}1} = (Q_{0,\,p\text{-}1} = (F_{0,0,\,p\text{-}1} = (H_{0,0,0,\,p\text{-}1} \| \dots H_{h\text{-}1,0,0,\,p\text{-}1}) \| \dots F_{s\text{-}1,0,\,p\text{-}1} = (H_{0,s\text{-}1,0,\,p\text{-}1} \| \dots H_{h\text{-}1,s\text{-}1,0,\,p\text{-}1})) \| \dots$$
$$Q_{q\text{-}1,\,p\text{-}1} = (F_{0,\,q\text{-}1,\,p\text{-}1} = (H_{0,0,\,q\text{-}1,\,p\text{-}1} \| \dots H_{h\text{-}1,0,\,q\text{-}1,\,p\text{-}1}) \| \dots F_{s\text{-}1,\,q\text{-}1,\,p\text{-}1} = (H_{0,s\text{-}1,\,q\text{-}1,\,p\text{-}1} \| \dots H_{h\text{-}1,s\text{-}1,\,q\text{-}1,\,p\text{-}1})))).$$

Формула утворена $2p+4q+6s+8h$ кількістю знаків.

Отже, формула модифікованої алгебри алгоритмів має на (2p+4q+6s+8h)/41 менше знаків.

3. Модель графічно-функціональної декомпозиції підсистеми ($\%P_k$), на візуальноелементну (W_k) та елементновластивістну (E_k) складові має вигляд:

$$\%P_k = \overline{W_k\,;\,E_k}\,,$$

де $\%$ – ідентифікатор підсистеми з інтерфейсом; k – змінна номера підсистеми, $k \in \{0, 1, ...\ K\text{-}1\}$. Вона має 10 знаків.

4. Описана системою алгоритмічних алгебр: $\%P_k = W_k \,\|\, E_k$ з 9 знаками.

Отже, формула модифікованої алгебри алгоритмів має на один знак більше, ніж формула модифікованої системи алгоритмічних алгебр.

5. *Візуальноелементна декомпозиція* починається заданням початкового (кореневого) візуального елемента і подальша його декомпозиція виконується за вибираними критеріями, наприклад, потрібної функціональності, вкладеності (котрі елементи поміщаються в інші), взаємного розташування елементів тощо. Формула візуальноелементної декомпозиції використовує операцію паралелення модифікованої алгебри алгоритмів:

$$\overline{W_k;\ \varnothing i_k\, \overline{R_{i,k};\ \varnothing j_{i,k}\, \overline{Q_{j,i,k}\,;\ \varnothing t_{j,i,k}\, S}}}_{t,j,i,k}\,,$$

де W_k – k-та візуальна складова; i_k – змінна вкладених $R_{i,k}$ елементів у елемент W_k; $i \in \{0, 1, ...\ I_k\text{-}1\}$; $j_{i,k}$ – змінна вкладених елементів $Q_{j,i,k}$ в елемент $R_{i,k}$; $j_{i,k} \in \{0, 1, ...\ J_{i,k}\text{-}1\}$; $t_{j,i,k}$ – змінна вкладених в елемент $Q_{j,i,k}$ елементів $S_{t,j,i,k}$; $t \in \{0, 1, ...\ T\text{-}1\}$.

6. *Візуальноелементна декомпозиція* є такою формулою модифікованої системи алгоритмічних алгебр:

$W_k * R_{0,k} * Q_{0,i,k} * S_{0,j,i,k} \| \dots S_{T\text{-}1,j,i,k} \| \dots Q_{J\text{-}1,i,k} * S_{0,j,i,k} \| \dots S_{T\text{-}1,j,i,k} \| \dots R_{I\text{-}1,k} * Q_{0,i,k} * S_{0,j,i,k} \| \dots S_{T\text{-}1,j,i,k} \| \dots Q_{J\text{-}1,i,k} * S_{0,j,i,k} \| \dots S_{T\text{-}1,j,i,k}$, з кількістю знаків $5I+7J+9T\text{-}1$.

Формула модифікованої алгебри алгоритмів має у ($5I+7J+9T$-1)/41 раз менше знаків.

7. *Елементновластивістна декомпозиція*, як і візуальноелементна, починається заданням кореневого візуального елемента і подальша його декомпозиція виконується за вибраними критеріями, наприклад, глобальними: пов’язання з іншими підсистемами, а також локальними: довжини і ширини, назви, вкладеності (котрі елементи поміщаються в інші):

$$E_k = \not\subset e_k E_{e,k};\ \varnothing x_{e,k}\ H_{x,e,k};\ \not\subset h_{e,x,k} H_{h,x,e,k};\ \varnothing y_{x,e,k}\ G_{y,x,e,k};\ \not\subset g_{y,x,e,k}\ G_{g,y,x,e,k},$$

де e_k – змінна властивостей елемента E_k, $e_k \in \{0, 1, \ldots A_{e,k}\text{-}1\}$; $E_{e,k}$ – властивості елемента E_k; $x_{e,k}$ – змінна вкладених описів елементів $H_{x,e,k}$ в елемент E_k; $h_{e,x,k}$ – змінна властивостей елемента $H_{x,e,k}$; $x_{e,k} \in \{0, 1, \ldots X_{e,k}\text{-}1\}$; $y_{x,e,k}$ – змінна вкладених описів елементів $G_{y,x,e,k}$ в елемент $H_{x,e,k}$, $y_{x,w,k} \in \{0, 1, \ldots Y_{x,e,k}\text{-}1\}$; $g_{y,x,e,k}$ – змінна властивостей елемента $G_{y,x,e,k}$, $g_{y,x,e,k} \in \{0, 1, \ldots G_{y,x,e,k}\text{-}1\}$.

8. Формула *елементновластивістної декомпозиції* абстрактної комп'ютерної системи у модифікованій системі алгоритмічних алгебр є такою:

$$E_k = \{[e_k]\ E_{e,k}\} * (H_{0,e,k} * \{[\ h_{x,e,k}]\ H_{h,x,e,k} * (G_{0,x,e,k} * \{[g_{0,x,e,k}]\ G_{g,0,x,e,k}\} \| \ldots$$
$$G_{Y\text{-}1,x,e,k} * \{[g_{Y\text{-}1,x,e,k}]\ G_{g,Y\text{-}1,x,e,k}\}) \| \ldots$$
$$H_{X\text{-}1,e,k} * \{[\ h_{X\text{-}1,e,k}]\ H_{h,X\text{-}1,e,k} * (G_{0,X\text{-}1,e,k} * \{[g_{0,x,e,k}]\ G_{g,0,X\text{-}1,e,k}\} \| \ldots$$
$$G_{Y\text{-}1,X\text{-}1,e,k} * \{[g_{Y\text{-}1,X\text{-}1,e,k}]\ G_{g,Y\text{-}1,X\text{-}1,e,k}\})).$$

Вона має $23X+29Y$ знаків.

Формула модифікованої алгебри алгоритмів має у (23X+29Y) /78 рази менше знаків.

9.*Функціональна декомпозиція* у модифікованій алгебрі алгоритмів є формулою:

$$F_r = \varnothing q_r x_{q,r}\ ;\ \varnothing v_r V_{v,r}\ ;\ \varnothing p_r M_{p,r}(h, \ldots g), \ldots \not\subset l_r L_{l,r}\ ,$$

де F_r – r-та функціональна підсистема; q_r – змінна циклу за кількістю змінних функціональної підсистеми F_r, $q_r \in \{0, 1, \ldots Q_k\text{-}1\}$ $x_{q,,r}$ – змінні; v_r – змінна циклу за кількістю властивостей, $v_r \in \{0, 1, \ldots N_k\text{-}1\}$; $V_{v,r}$ – властивість; p_r – змінна циклу за кількістю процедур, $v_r \in \{0, 1, \ldots P_k\text{-}1\}$; $M_{p,r}(h, \ldots g)$ – процедура, яка залежна від змінних $h, \ldots g$; l_r – змінна циклу за кількістю інших складових; $L_{l,r}$ – інші складові функціональної підсистеми. У цій формулі є 42 знаки.

10. *Функціональна декомпозиція у* модифікованій системі алгоритмічних алгебр є виразом:

$F_r = (x_{0,,r} \| \ldots x_{Q,,r})*\ (V_{0,r} \| \ldots V_{N,r})*(M_{0,r}(h, \ldots g) \| \ldots M_{P,r}(h, \ldots g))* \ldots \{[l_r]\ L_{l,r}\}$.

Нараховує він $5(Q+N)+10P$ кількість знаків.

Тож формула модифікованої алгебри алгоритмів має у $(5(Q+N)+10P)/42$ раз менше знаків.

10. Процедури з виключеннями є такою формулою модифікованої алгебри алгоритмів:

$$M_{p,r}(h, \ldots g) = \not\subset d_{p,r} n_{d.p,r}, \not\supset b_{p.r} C_{b,p,r}\ , \ldots \not\subset x_{p,r} S_{x,p,r}\ ,$$

де $M_{p,r}(h,\ldots g)$ – процедура, залежна від змінних $h, \ldots g$; $d_{p,r}$ – змінна циклу за кількістю змінних процедури, $d \in \{0, 1, \ldots D_{p,r}\text{-}1\}$; $n_{d.p,r}$ – змінні процедури; $C_{b,p,,r}$ – елемент виключення; $b_{p,r}$ – змінна циклу за кількістю виключень $b \in \{0, 1, \ldots B_{b,p,r}\text{-}1\}$; $x_{p,r}$ – змінна циклу за кількістю інших складових; $S_{x,p,r}$ – інші складові функціональної підсистеми. Має формула 45 знаків.

11. Функціональна декомпозиція у модифікованій системі алгоритмічних алгебр є таким виразом:

$M_{p,r}(h, \ldots g) = \{[d_{p,r}]\ n_{d.p,r}\}*([b=0]\ C_{0,p,r}, ([b=1]\ C_{1,p,r},, \ldots ([b= B_{p,r}\text{-}1]\ C_{0,p,r},,) \ldots))* \ldots \{[x_{p,r}]\ S_{x,p,r}\}$,

який має $14B$ знаків. Звідси опис модифікованою алгеброю алгоритмів має у $(40+14 \times B_{b,p,r})/45$ раз менше знаків.

12. Декомпозиція *процедур з циклічними паралеленнями* є такою формулою модифікованої алгебри алгоритмів:

$$M_{p,r}(h, \ldots g) = \not\subset d_{p,r} n_{d.p,r}\ , \varnothing b_{p.r} C_{b,p,r}\ , \ldots \not\subset x_{p,r} S_{x,p,r}\ ,$$

де $M_{p,r}(h,\ldots g)$ – процедура, яка залежна від змінних $h, \ldots g$; $d_{p,r}$ – змінна циклу за кількістю змінних процедури, $d \in \{0, 1, \ldots D_{p,r}\text{-}1\}$; $n_{d.p,r}$ – змінні процедури; $C_{b,p,,r}$ – елемент виключення; $b_{p,r}$ – змінна циклу за кількістю виключень $b \in \{0, 1, \ldots B_{b,p,r}\text{-}1\}$; $x_{p,r}$ – змінна циклу за кількістю інших складових; $S_{x,p,r}$ – інші складові функціональної підсистеми. Формула містить 45 знаків.

13. Модифікована система алгоритмічних алгебр процедури з паралеленнями має такий вигляд:

$$M_{p,r}(h, \ldots g) = \{[d_{p,r}]\ n_{d.p,r}\}*(C_{0,p,r} \| C_{1,p,r} \| \ldots C_{B\text{-}1,p,r})* \ldots \{[x_{p,r}]\ S_{x,p,r}\}$$

і нараховує 7*B* знаків. Отож, формула модифікованої алгебри алгоритмів має у $7B_{b,p,r}/45$ раз менше знаків.

У табл. 4 наведено результати порівнянь моделей абстрактної системи, описаних модифікованою системою алгоритмічних алгебр і модифікованою алгеброю алгоритмів.

Таблиця 4

Порівняння моделей математичного забезпечення підсистем абстрактної системи

№	Модель декомпозиції складової абстрактної системи	Система алгоритмічних алгебр модиф. (кіль-ть знаків)	Модифікована алгебра алгоритмів (кіль-ть знаків)	Зменшено кількість знаків (у рази)
1	*Підсистемна*	$2p+4q+6s+8h$	4+7+11+15	$(2p+4q+6s+8h)/37$
2	*Графічно-функціональна*	9	10	–
3	*Візуальноелементна*	$5I+7J+9T-1$	7+11+15	$(5I+7J+9T-1)/33$
4	*Елементновластивістна*	$23X+29Y$	26+34	$(23X+29Y)$ /60
5	*Функціональна*	$5(Q+N)+10P$	7+7+12	$(5(Q+N)+10P)/26$
6	*Процедури з виключеннями*	$14B$	11	$14B/11$
7	*Процедури з паралеленнями*	$7B$	11	$7B/11$

Висновки

1. Модифікованою алгеброю алгоритмів описується як клас асоціативних, так і клас неасоціативних алгоритмів.
2. Формули модифікованої алгебри алгоритмів мають меншу кількість знаків, ніж формули алгебри алгоритмів.
3. Моделі комп'ютерних систем модифікованою алгеброю алгоритмів порівняно з моделями модифікованої системи алгоритмічних алгебр описуються меншою кількістю знаків.
4. Розроблену математичну модель узагальнених абстрактних інструментальних засобів з інтерфейсом користувача можна застосувати для синтезу моделей прикладних інструментальних засобів.

1. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра. Языки. Программирование. – 2-е изд., перераб. – К.: Наукова думка, 1978. – 318 с. 2. Цейтлин Г.Е. Введение в алгоритмику. – К. : Сфера, 1998. – 310 с. 3. Ovsyak A. The extended algebra of algorithms width multiconditional elimination / V. Ovsyak, A. Ovsyak // Вісник Національного університету "Львівська політехніка": Комп'ютерні науки та інформаційні технології. – 2010. – № 672. – С. 291 – 300. 4. Овсяк О. Розширена алгебра алгоритмів і модель зв'язку між класичною і розширеною операцією елімінування / О. Овсяк // Збірник наукових праць „Комп'ютерні технології друкарства". – № 23. 2010. – С.45–53. 5. Owsiak W., Owsiak A. Rozszrzerzenie algebry algorytmów // Pomiary, automatyka, kontrola. – № 2, 2010. – S. 184–188. 6. Математика: Большой энциклопедический словарь / гл. ред. Ю.В. Прохоров. – М.: Большая рос. энц. 1998. – 848 с. 7. Бородин А.И., Бугай А.С. Выдающиеся математики: биогр. слов. -справ. – К.: Рад. шк., 1987. – 656 с. 8. Шеренга великих математиков / под ред. проф. Влод. Крысицкого. – Варшава: Наша ксенгарня, 1970. – 188 с. 9. Драґан Я., Овсяк В. Системний аналіз і методологія алгебри алгоритмів // Вісник Нац. ун-ту "Львівська політехніка". Комп'ютерні науки та інформаційні технології. – 2012. – № 732, – С.91–95. 10. Драґан Я. Межа Бремермана, системний аналіз і антологія складності моделей соціотехнічних систем // Вісник Нац. ун-ту "Львівська політехніка" Комп'ютерні науки та інформаційні технології. – 2010. –

№ 663. – С.78–83. 11. Драган Я., Медиковський М., Овсяк В., Сікора Л., Яворський Б. Системний аналіз концепції та принципів побудови математичної моделі досліджуваного об'єкта в фізико-технічних науках та оцінювання її якості // Вісник Нац. ун-ту "Львівська політехніка" Комп'ютерні науки та інформаційні технології. – 2010. – № 686. – С.170–179. 12. Биркгоф Г. Математика и психология. – М.: Сов. радио, 1977. – 96 с. 13. Детловс В. Эквивалентность нормальных алгоритмов и рекурсивных функций // Тр. матем. ин-та АН СССР им. В.А. Стеклова, 1958, 52. – С.75–139. 14. Адамар Ж. Исследование процесса изобретения в области математики / пер. с франц. М.А. Шаталовой и О.П. Шаталова, под ред. И.Б. Погребысского. – М.: Сов. радио, 1970. – 152 с. 15. Ovsiak V.K. Computational models and algebra of algorithms // Вісник Нац. ун-ту "Львівська політехніка" Інформаційні системи та мережі. – 2008. – № 621. – С.3–18.

УДК 004.9

Т. Свірідова, У. Марікуца
Національний університет "Львівська політехніка",
кафедра систем автоматизованого проектування

ДОСЛІДНИЦЬКЕ ТЕСТУВАННЯ ЯК ЗАСІБ ПІДВИЩЕННЯ ЯКОСТІ АПЛІКАЦІЇ

Запропоновано використання дослідницького тестування як спосіб покращення якості аплікації, що розробляється. Розглянуто дослідницьке тестування як частину регресійного тестування.

Ключові слова: дослідницьке тестування, регресійне тестування, якість.

This paper is devoted to the investigation of quality factors which can influence on developed software. Exploratory testing is described and analysed in scope of regression testing.

Key words: exploratory testing, regression testing, quality.

Вступ

Однією з умов успішності розробленого програмного продукту є його якість. Якість програмного продукту – це сукупність його рис і характеристик, які впливають на його здатність задовольняти задані потреби користувачів.

Як відомо, коли розробник додає нову функціональність або виправляє дефект у системі, існує висока ймовірність того, що внесені зміни можуть повністю дестабілізувати систему. Якщо під час кожної зміни коду тестувати тільки новий функціонал, то не можна бути повністю впевненим, що всі решта складові системи працюють відповідно до вимог або очікувань клієнта. Власне для переконання, що аплікація цілісна та працює повністю правильно, розроблено такий підхід для тестування, як регресія. Однак, як і будь-який інший підхід, регресія має низку недоліків, одними з яких є так званий "ефект пестициду", одноманітність роботи, прогін тих самих не завжди актуальних, що, як результат, призводить до зниження якості кінцевої версії аплікації. Для покращення процесу регресійного тестування запропоновано використати дослідницьке тестування як його частину.

Основна частина

Дослідницьке тестування – це розроблення та виконання тестів у той самий час, що є повною протилежністю підходу з тест-кейсами. Дослідні тести, на відміну від класичних тестів, не визначені заздалегідь і не виконуються точно відповідно до плану.