УДК 004.021

А.С. Василюк, Т.М. Басюк
Національний університет "Львівська політехніка",
кафедра інформаційних систем та мереж

АЛҐОРИТМИ ЗГОРТАННЯ І РОЗГОРТАННЯ ФОРМУЛ АБСТРАКТНИХ АЛҐОРИТМІВ

Подано означення процесів згортання, розгортання й адаптації формул алґоритмів. Наведено алґоритм комп'ютерної адаптації формул алґоритмів. Синтезовано, мінімізовано, побудовано математичну модель і досліджено алґоритм адаптації базового знака операції. Наведено моделі алґоритмів згортання і розгортання формул алґоритмів. Показана верифікація розроблених алґоритмів.

Ключові слова – процеси згортання, розгортання і адаптації формул алґоритмів.

Determination of processes of rolling up, development and adaptation of formulas of algorithms is described. The algorithm of adaptation of formulas of algorithms is resulted. It is synthesized, it is minimized, a mathematical model and probed algorithm of adaptation of base sign of operation is built. The models of algorithms of rolling up and development of formulas of algorithms are resulted.

Keywords – processes of rolling up, development and adaptation of formulas of algorithms.

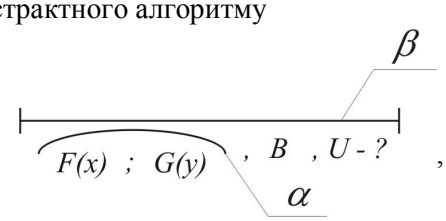
Вступ

Відома [1, 2] алгебра алґоритмів, яка має оригінальні операції, наприклад, секвентування, елімінування, паралелення та циклічні операції, котрі позначаються спеціальними знаками, яких немає серед відомих математичних знаків. Для набору та редагування формул абстрактних алґоритмів розроблено спеціалізовану комп'ютерну підсистему МОДАЛ [3]. Але вона не виконує такі операції, як автоматична адаптація, згортання та розгортання формул абстрактних алґоритмів.

У таких мовах програмування, як Microsoft Visual C#. Net, Borland Delphi 8.0 та інших поширена технологія згортання та розгортання підпрограм. Завдяки такій можливості значно зменшується видимий код написаної програми. Це особливо актуально, коли код програми громіздкий. Але така проблема є не тільки в мовах програмування. Вона не втрачає актуальності і при створенні алґоритмів, адже оперувати великими алґоритмами також нелегко. Використовуючи згорнуті формули, набагато зручніше створювати алґоритми, зменшується кількість механічних помилок оператора набору. Тому у цій статті розглядається проблема згортання та розгортання формул абстрактних алґоритмів.

Ілюстрація згортання та розгортання формул абстрактних алґоритмів

Нехай дано формулу абстрактного алгоритму



, (1)

де α – знак операції секвентування, β – знак операції елімінування, *F(x), G(y), B, U-?* – унітерми.

Унітермами є вирази мови, зокрема математичної мови [4].

Унітерми *F(x), G(y)* є *вкладеними* в знак операції секвентування – *α*. Відносно цих унітермів знак операції секвентування не є вкладеним. Назвемо його *базовим.* Кожен з унітермів чи знаків операцій має певні параметри.

Такими параметрами є: *id* – унікальний ідентифікаційний номер формули, *TypeObj* – тип формули, *index_op_a* – ідентифікаційний номер першої вкладеної формули, *index_op_b* – ідентифікаційний номер другої вкладеної формули, *FormulaName* – назва формули, *Text* – текст терма [5].

$$\vdash \overline{\#\_\alpha \ , \ B \ , U - ?} \ , \tag{2}$$

де #_ *α* – умовне позначення формули алґоритму.

Згортання – процес перетворення формули алґоритму (формула *α* в формулі (1)), яка складається з елементарних термів до її умовного позначення (наприклад, унітерм #_ *α* у формулі (2)).

Розгортання – процес перетворення умовного позначення формули алґоритму (наприклад, унітерм #_ *α* в формулі (2)) до формули алґоритму (1).

Адаптацією називатимемо процес переміщення вкладених формул відносно базового знака операції і зміни його розмірів відносно вкладених формул.

Алґоритм комп'ютерної адаптації формул алґоритмів

Виконуючи операції згортання та розгортання, необхідно автоматично адаптувати положення вкладених формул і розміри базових знаків операцій. Загальний алґоритм адаптації описується такою формулою:

$$\left(\begin{array}{l} l = 0 \\ ; \\ \left(\begin{array}{l} K \\ ; \\ \varnothing \ i = n1 \\ \vdash \overline{* \ , \ c_i \ , (i = n1) - ?} \end{array} \right. \end{array} \right. \ ,$$

де *l=0* – ініціалізація автоматичної адаптації формули абстрактного алґоритму, *K* – алґоритм адаптації базової формули абстрактного алґоритму, n_1 – максимальна кількість знаків операцій та термів абстрактного алґоритму.

Синтез абстрактного алґоритму адаптації базового знака операції

Для синтезу абстрактних алґоритмів розроблено два методи [1, 2]: секвенційний та табличний. Секвенційний метод є простішим, тому синтез абстрактного алґоритму адаптації виконано власне цим методом. Він передбачає виконання синтезу абстрактного алґоритму за два етапи. На першому етапі синтезуються секвенції, а на другому – елімінування.

Перший етап – синтез секвенцій.

Абстрактний алґоритм адаптації базового знака операції передбачає такі секвенції. Перша секвенція відповідає за адаптацію першої вкладеної формули до базової формули і адаптацію базової формули до вкладеної. Назвемо її S_1. Друга секвенція відповідає за адаптацію другої вкладеної формули до базової формули та адаптацію базової формули до вкладеної. Назвемо її S_2. Третя секвенція описує адаптацію базової формули, коли немає першої та другої вкладених в базову формулу формул. Назвемо її S_3.

Наступні три секвенції, на відміну від попередніх трьох, описують аналогічні процеси, тільки за умови вертикальної орієнтації знака операції. Орієнтація знака операції може бути горизонтальною або вертикальною.

Для синтезу секвенцій S_1, S_2, S_3, S_4, S_5, S_6 нам необхідно описати такі унітерми: $P(k,0)$ – ініціалізація алґоритму адаптації формули абстрактного алґоритму, $P(k, a)$, $P(k, a')$ – терми адаптації першої вкладеної формули в базову залежно від типу орієнтації знака операції, $P(k, b)$, $P(k, b')$ – терми адаптації другої вкладеної формули в базову залежно від типу орієнтації знака операції, $K_1(x, y)$, $K_1(x', y')$ – унітерми ідентифікації підалґоритмів адаптації першої та другої вкладених формул в базову залежно від орієнтації знака операції, K_2, K_2' – унітерми ідентифікації підалґоритмів адаптації базової формули залежно від вкладених формул

$$S_1 = \Big(\big(P(k,0)\ ;\ P(k,a)\ ;\ K1(x,y)\ ;\ K2\big) \qquad S_2 = \Big(\big(P(k,0)\ ;\ P(k,b)\ ;\ K1(x,y)\ ;\ K2\big) \qquad S_3 = \big(P(k,0)\ ;\ K2\big)$$

$$S_4 = \Big(\big(P(k,0)\ ;\ P(k,a')\ ;\ K1(x',y')\ ;\ K2'\big) \qquad S_5 = \Big(\big(P(k,0)\ ;\ P(k,b')\ ;\ K1(x',y')\ ;\ K2'\big) \qquad S_6 = \big(P(k,0)\ ;\ K2'\big) \tag{3}$$

Другий етап – синтез елімінувань.

Секвенції S_1 і S_2 елімінуємо за умовою наявності другої вкладеної формули (унітерм U1(b, c1) – ?). Отримане елімінування із секвенцією S_1 елімінуємо за умовою наявності першої вкладеної формули (унітерм U1(a, c1) -?). Секвенції S_5 і S_6 елімінуємо за умовою наявності другої вкладеної формули (унітерм U1(b', c1) -?). Одержане елімінування із секвенцією S_4 елімінуємо за умовою наявності першої вкладеної формули (унітерм U1(a', c1) -?).

Для синтезу елімінування L_1 необхідно описати такі унітерми: S_1, S_2 – секвенції з формули (3), U1(b, c1) -? – умовний унітерм перевірки на наявність другої вкладеної формули. Аналогічно до того, як синтезується елімінування L_1, синтезуються всі решта елімінувань. У результаті виконаних дій отримуємо такі елімінування:

$$L_1 = \vdash S_2\ ,\ S_3\ ,\ U1(b,c1) - ? \qquad L_3 = \vdash S_5\ ,\ S_6\ ,\ U1(b',c1) - ?$$

$$L_2 = \vdash S_1\ ,\ L_1\ ,\ U1(a,c1) - ? \qquad L_4 = \vdash S_4\ ,\ L_3\ ,\ U1(a',c1) - ? \tag{4}$$

$$L_5 = \vdash L_2\ ,\ L_4\ ,\ U1(g,c1) - ?$$

Мінімізація абстрактного алгоритму

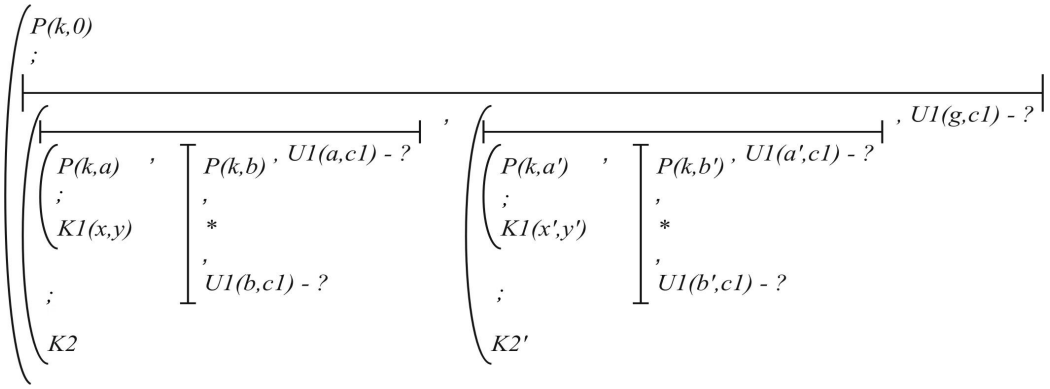
Підставимо в елімінування L_1 (4) секвенції S_2 та S_3 (3).

На основі властивості виносу унітерму за знак операції елімінування [3] виносимо унітерми $P(k,0)$ і K_2 за знак операції елімінування й отримаємо таку формулу:

$$\vdash \Big(\big(P(k,0)\ ;\ P(k,b)\ ;\ K1(x,y)\ ;\ K2\big)\ ,\ \big(P(k,0)\ ;\ K2\big)\ ,\ U1(b,c1) - ?$$

$$\Big(P(k,0)\ ;\ \vdash \big(P(k,b)\ ,\ *\ ,\ U1(b,c1) - ?\ ;\ K1(x,y)\ ;\ K2\big)\Big)$$

Аналогічно мінімізуються всі решта елімінувань. Виконавши підстановку та мінімізацію, одержимо таку формулу:



Модель абстрактного алґоритму

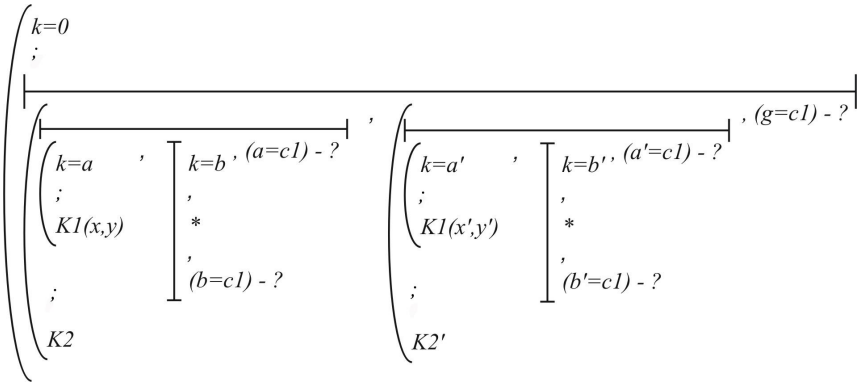
Модель абстрактного алґоритму будують, замінивши абстрактні унітерми предметними (конкретними) і задавши секвентні області значень змінних та унітермів [1, 2].

Замінюємо абстрактні унітерми на предметні і заданням секвентних областей значень змінних й унітермів будуємо модель абстрактного алґоритму.

Абстрактний двомісний унітерм $P(m, n)$ замінюємо на предметний унітерм присвоєння $m=n$. Тоді $P(k,0)$ замінюється на $k=0$, $P(k, a)$ на $k=a$, $P(k, b)$ на $k=b$, $P(k, a')$ на $k=a'$, $P(k, b')$ на $k=b'$. Абстрактний умовний терм $U_1(p, q)$-? замінюємо предметним термом порівняння $(p=q)$-?. У такому разі унітерм перевірки орієнтації виразу $U_1(g, c_1)$-? замінюємо на $(g=c_1)$-?, унітерм перевірки на наявність першого вкладеного виразу в двомісну операцію $U_1(a, c_1)$-? на $(a=c_1)$-?, $U_1(b, c_1)$-? на $(b=c_1)$-?, $U_1(a', c_1)$-? на $(a'=c_1)$-?, $U_1(b', c_1)$-? на $(b'=c_1)$-?. Для c_1 задаємо таке значення: $c_1=0$.

Абстрактні унітерми $K_1(x, y)$, $K_1(x', y')$, K_2 і K_2' є складними абстрактними виразами, які, враховуючи обмежений обсяг статті, не можуть бути розкриті у ній. Моделі цих абстрактних унітермів будуються аналогічно, як це було продемонстровано вище.

Отримана модель абстрактного алґоритму матиме такий вигляд:



Дослідження моделі алґоритму

Теорема. Математичною моделлю алґоритму описуються адаптація вкладених формул до знаків базових операцій та ідентифікація алґоритмів адаптації знаків базових операцій до вкладених формул.

Доведення. Для доведення використовуємо метод трансфінітної математичної індукції [4], оскільки доведення за всіх можливих значень змінних є аналогічним, а самі значення змінних є впорядкованими.

Доведемо твердження для змінної a за таких значень всіх решти змінних $a'=3$, $b=4$, $b'=5$, $g=0$. Нехай твердження справедливе за будь-яких $i < p$, де $p < n1$. При $a = p$ змінній k присвоюється значення 0. З елімінування за умовою $(g=c_1)$-?, якою ідентифікується горизонтальна конфігурація

операції, для $g = 0$ умова $(0 = 0)$-? виконується, отримуємо елімінування за умовою $(a \neq c_1)$-?, якою ідентифікується перший вираз двомісної операції теорії абстрактних алґоритмів. При $a = p$ одержимо $(p \neq 0)$-?. Якщо $p \neq 0$, то умова $(0 \neq 0)$-? не виконується, тоді виконується предметний терм присвоєння $k = a$, яким ідентифікується перша формула двомісної операції.

Далі термом $K_1(x, y)$ описується ідентифікація підалґоритмів адаптації першої формули, вкладеної в базовий.

Далі йде унітерм K_2, яким описується ідентифікація підалґоритму адаптації базових операцій до першої формули двомісної операції.

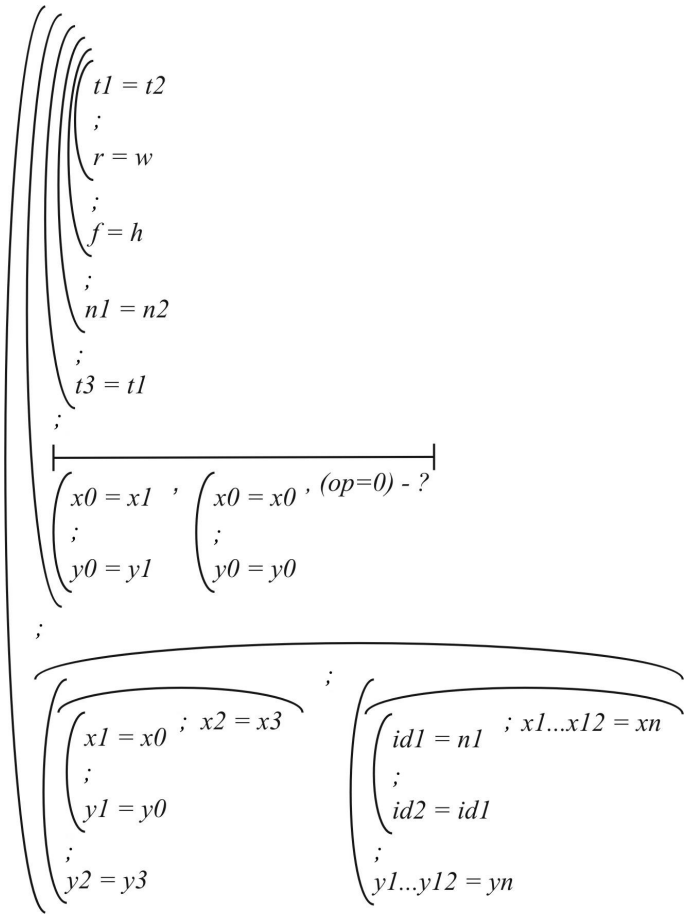
Тепер, якщо $p \neq 0$, то умовним термом $(b \neq c_1)$-? описується ідентифікація другого виразу двомісної операції. За заданих значень $(b = 4)$ умовний терм $(4 \neq 0)$-? виконується, тоді виконується терм ідентифікації другого вкладеного виразу в базовий. Термом K_2 описується ідентифікація підалґоритму адаптації базової операції до другої вкладеної формули.

Аналогічно теорема доводиться для всіх решти значень змінних. Теорему доведено.

Моделі абстрактних алгоритмів згортання та розгортання формул абстрактних алґоритмів

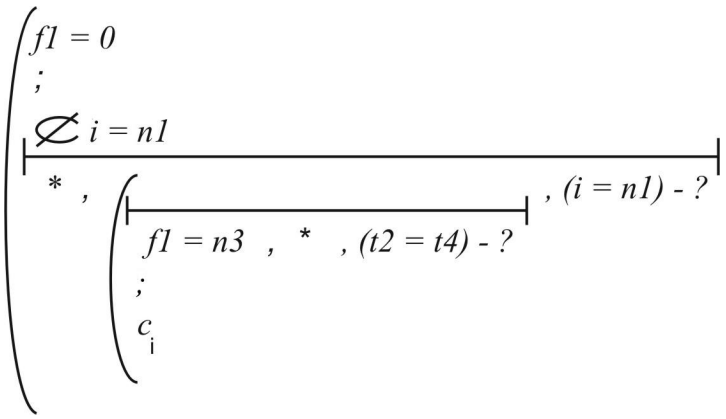
Модель абстрактного алґоритму згортання формул абстрактних алґоритмів

Наступна формула абстрактного алґоритму є моделлю абстрактного алґоритму згортання формул абстрактних алґоритмів. Для створення цієї моделі необхідно мати такі унітерми: $t1 = t2$ – унітерм присвоєння назви згорнутої формули, $r = w$ – унітерм визначення ширини згорнутої формули, $f = h$ – унітерм визначення висоти згорнутої формули, $n1 = n2$ – збільшення максимальної кількості виразів в алґоритмі на 1, $t3 = t1$ – унітерм присвоєння назви згорнутої формули, $x0 = x1$, $y0 = y1$, $x0 = x0$, $y0 = y0$ – унітерми збереження координат нової формули, $(0p = 0)$-? – умовний двомісний унітерм вибору формули, $x1 = x0$, $y1 = y0$, $x2 = x3$, $y2 = y3$ – унітерми визначення координат для згорнутої формули, $id1 = n1$, $id2 = id1$ – унітерми створення нової формули, $x1 \ldots x12 = xn$, $y1 \ldots y12 = yn$ – унітерми визначення координат переміщення розгорнутої формули в невидиму область.



Модель абстрактного алґоритму розгортання формул абстрактних алґоритмів

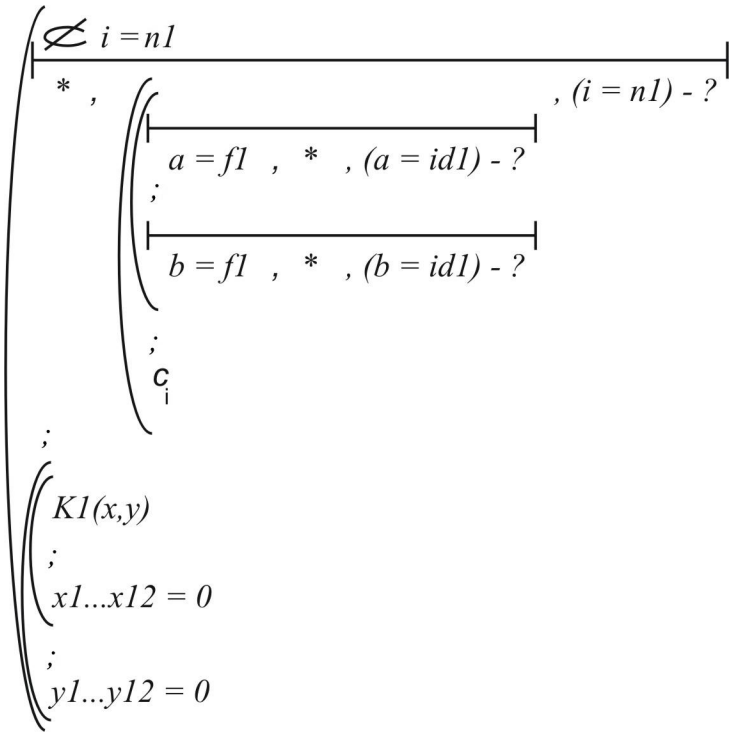
Для створення моделі абстрактного алґоритму розгортання формул абстрактних алґоритмів необхідно спочатку описати модель пошуку формули за назвою згорнутої формули.



Модель описує процес пошуку формули за його назвою. Вона містить такі унітерми: *f1 = 0* – ініціалізації системи пошуку за назвою, *f1 = n3* – унітерм присвоєння критерію пошуку, *(t2 = t4) –?* – умовний унітерм перевірки на наявність шуканої формули, *(i = n1) -?* – умовний унітерм закінчення пошуку формули в формулі абстрактного алґоритму.

Використовуючи вищеописану модель пошуку виразу за назвою, створюємо модель абстрактного алґоритму розгортання формул абстрактних алґоритмів.

Формула абстрактного алґоритму (11) є моделлю абстрактного алґоритму розгортання формул абстрактних алґоритмів. Для створення цієї моделі використано такі унітерми: *a = f1* – ініціалізація згорнутої першої вкладеної формули, *(a = id1) -?* – унітерм перевірки на наявність першої вкладеної формули, *(i = n1) -?* – умовний унітерм закінчення розгортання, *b = f1* – ініціалізація згорнутої другої вкладеної формули, *(b = id1) -?* – унітерм перевірки на наявність другої вкладеної формули, *K1(x, y)* – унітерми ідентифікації підалґоритмів адаптації першої та другої вкладених формул в базовий, *x1…x12 = 0, y1…y12 = 0* – унітерм знищення згорнутої формули абстрактного алґоритму.



Верифікація розроблених моделей

На основі синтезованих та мінімізованих моделей згортання і розгортання формул абстрактних алґоритмів створено систему АбстрактАл, інтерфейс якої зображено нижче.

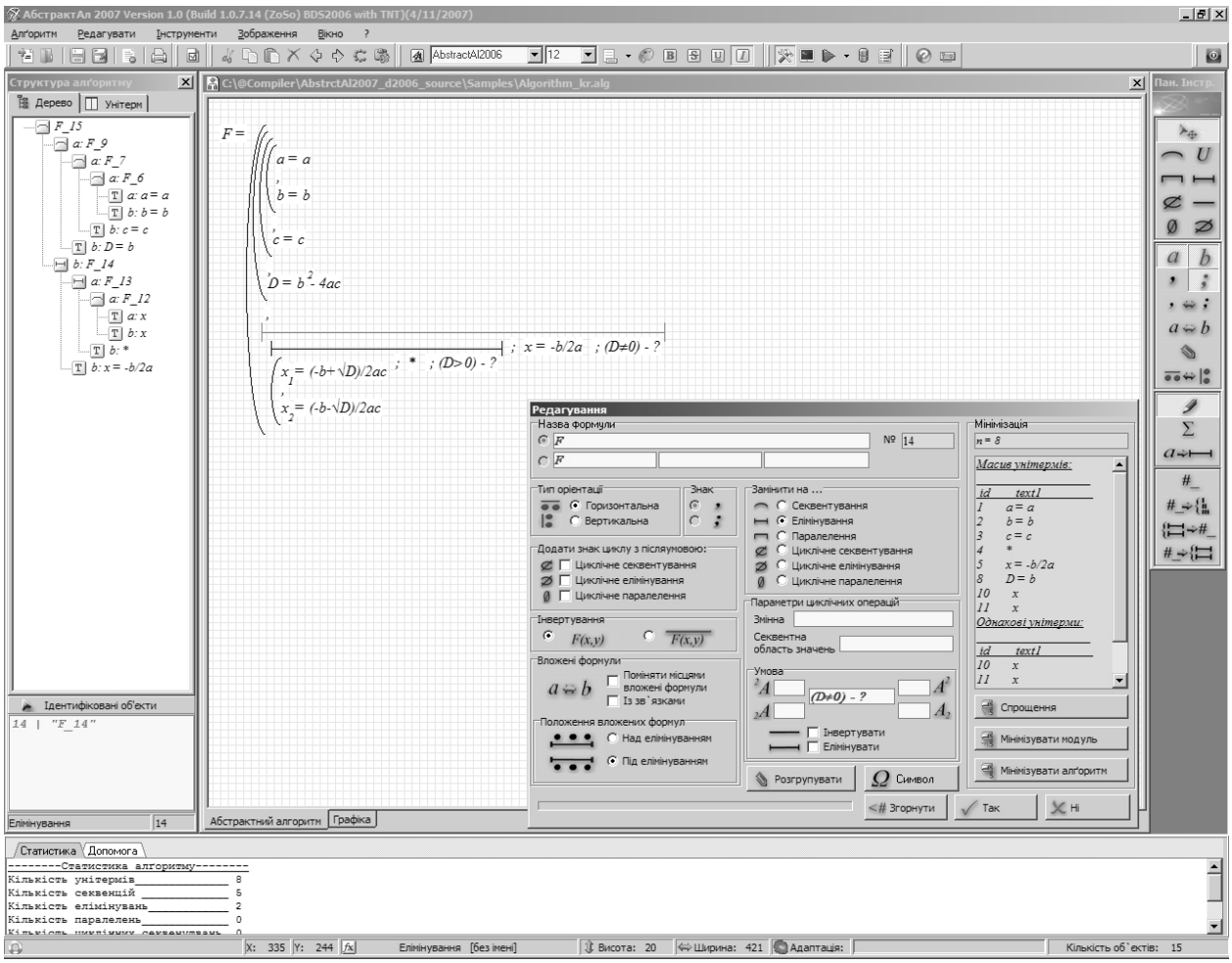


Рис. 1. Абстрактний алґоритм

На рис. 1 зображено реалізацію системи АбстрактАл, виконану засобами мови програмування Borland Delphi 7.0 [6, 7]. Ліворуч зображено формулу абстрактного алґоритму в розгорнутому вигляді, а праворуч зображено формулу абстрактного алґоритму після виконання операції згортання. Після використання операції часткового згортання (згортання вибраної формули алґоритму, а не всієї формули) формул абстрактних алґоритмів формула абстрактного алґоритму Евкліда стала лаконічнішою, читабельною і розмір формули алґоритму значно зменшився. Також важливим є те, що при повторенні згорнутої формули нема необхідності описувати всю формулу, а тільки її згорнутий вигляд.

Розглянемо приклад. Нехай необхідно згорнути, а потім розгорнути знак операції елімінування.

Дії, необхідні для вищеописаних операцій з алгоритмом:

- ідентифікувати знак операції елімінування;
- інструментом “Згорнути чи розгорнути формулу” здійснити згортання знака операції елімінування.

Результатом вищеописаних дій буде формула, відображена в головному вікні редактора:

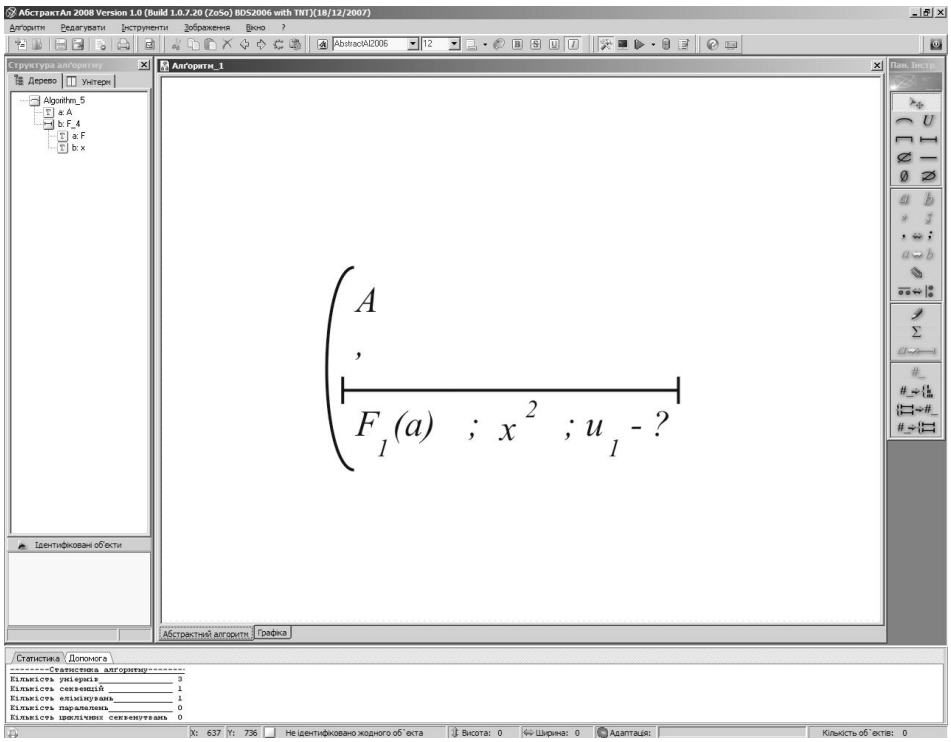


Рис. 2. Формула абстрактного алгоритму до процесу згортання

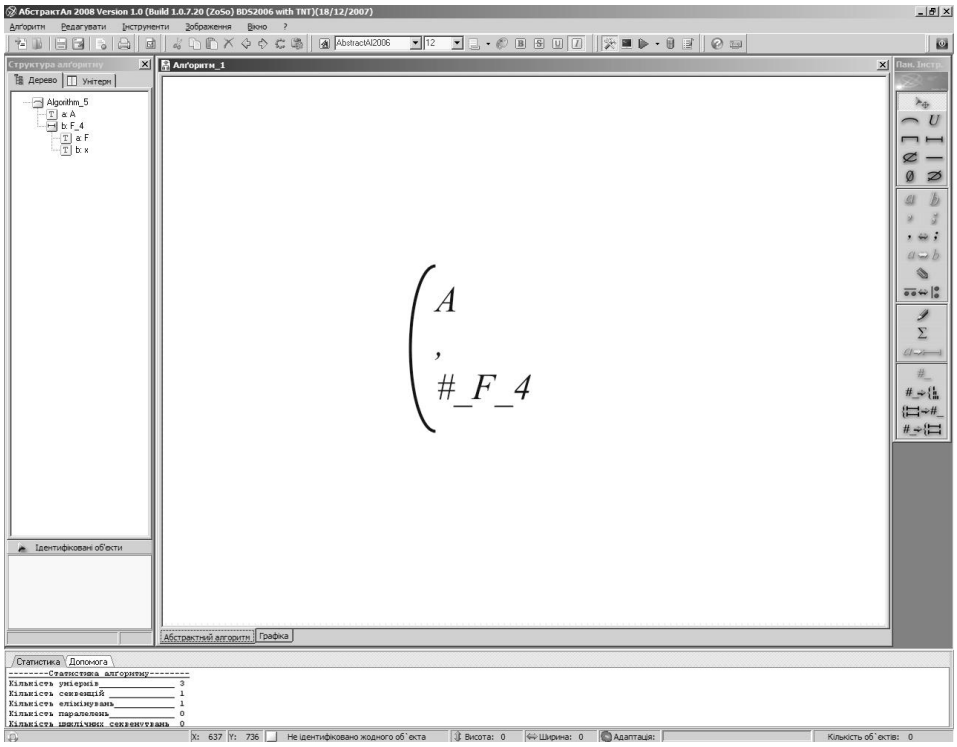


Рис. 3. Формула абстрактного алгоритму після процесу згортання

Дії, які необхідно здійснити для розгортання формули алгоритму:

- ідентифікувати згорнуту формулу;
- інструментом “Згорнути чи розгорнути формулу” здійснити розгортання згорнутої формули.

Результат роботи системи зображено на рис. 2 і 3.

Висновки

1. Синтезована, мінімізована і досліджена математична модель алґоритму адаптації формул абстрактних алґоритмів описує ідентифікацію підалґоритмів адаптації базових операцій до вкладених формул і вкладених формул до базових, реалізація якої забезпечує більшу наочність подання алґоритмів у вигляді формул теорії абстрактних алґоритмів.
2. Дослідження математичної моделі ще до її практичної реалізації та апробації забезпечило виявлення помилок, допущених під час її синтезу та доводить, що вона описує необхідні процеси.
3. Абстрактним алґоритмом адаптації формул абстрактних алґоритмів описано процеси автоматичного згортання та розгортання формул абстрактних алґоритмів.
4. Верифікацією розроблених математичних моделей алґоритмів підтверджена їх коректність. Програмна реалізація, крім автоматизованих процесів набору та редагування в автоматичному режимі, виконує адаптацію вкладених формул до базових і базових формул до вкладених.

1. Овсяк В., Бритковський В., Овсяк О., Овсяк Ю. Синтез і дослідження алґоритмів комп'ютерних систем. – Львів, 2004. – 276 с. 2. Овсяк В. АЛҐОРИТМИ: методи побудови, оптимізації, дослідження вірогідності. – Львів: Світ, 2001. – 160 с. 3. Бритковський В.М. Моделювання редактора формул секвенційних алґоритмів. Автореф. дис. роб. к.т.н. – Львів: видавничо-поліграфічний відділ ЛвЦНТЕІ. – 18 с. 4. Математическая энциклопедия: Гл. ред. И.М. Виноградов, т.3. – М.: Советская Энциклопедия, 1982. – 1184 с. 5. Овсяк В., Василюк А. Принцип побудови підсистеми редагування формул абстрактних алґоритмів // Комп'ютерні технології друкарства – Львів: УАД, – 2004. – № 12. – С. 137–146. 6. Архангельский А.Я. Delphi 6. Справочное пособие – М.: ЗАО "Издательство БИНОМ", 2001. – 1024 с. 7. Архангельский А.Я. Программирование в Delphi 6. – М.: ЗАО "Издательство БИНОМ", 2001.