

лише найімовірніших елементів, а ймовірності решти елементів моделювати за допомогою показникової чи рівномірної функції.

4. Прискорити декодування арифметичних кодів зі статичним формуванням інтервалів більш ніж на 40 % дає змогу використання допоміжного масиву, в якому для кожного значення загального інтервалу зберігається номер елемента, що йому відповідає.

Надалі, з метою подальшого зменшення КС в процесі прогресуючого ієрархічного стиснення зображень без втрат, нами планується розробити способи попереднього зменшення ентропії та пристосувати контекстно-залежні методи компресії для цього способу обходу пікселів.

1. Миано Дж. *Форматы и алгоритмы сжатия изображений в действии: учеб. пособ.* / Дж. Миано. – М. : Триумф, 2003. – С. 249-318. – (Практика программирования). 2. Сэломон Д. *Сжатие данных, изображений и звука* / Д. Сэломон. – М.: Техносфера, 2006. – 368 с. – (Мир программирования: цифровая обработка сигналов). 3. Бредихин Д. Ю. *Сжатие графики без потерь качества [Электронный ресурс]* / Д. Ю. Бредихин. – 2004. – [http://www.compression.ru/download/articles/i\\_lossless/bredikhin\\_2004\\_lossless\\_image\\_compression\\_doc.rar](http://www.compression.ru/download/articles/i_lossless/bredikhin_2004_lossless_image_compression_doc.rar). 4. *Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео* / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. – М. : ДИАЛОГ-МИФИ, 2003. – 384 с. 5. Гонсалес Р. *Цифровая обработка изображений* / Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2005. – 1072 с. 6. Прэтт Э. *Цифровая обработка изображений: Пер. с англ.* / Э. Прэтт. – М.: Мир, 1982. – Кн. 2, 480 с., ил. 7. Шпортько О. В. *Використання предикторів в процесі прогресуючого ієрархічного контекстно-незалежного стиснення зображень без втрат* / О. В. Шпортько // *Вісник Національного університету "Львівська політехніка"*. – 2013. – № 771. – С. 354–364. – (Серія: Комп'ютерні науки та інформаційні технології).

УДК 004.652

А.С. Василюк, Т.М. Басюк

Національний університет "Львівська політехніка",  
кафедра інформаційних систем та мереж

## ПІДСИСТЕМА ЗНИЩЕННЯ ФОРМУЛ АЛГОРИТМІВ

© Василюк А.С., Басюк Т.М., 2014

**Описано означення процесів знищення формул алгоритмів. Наведено алгоритм комп'ютерного знищення формул алгоритмів. Синтезовано, мінімізовано, побудовано математичну модель і досліджено алгоритм знищення формул абстрактних алгоритмів.**

**Ключові слова:** знищення, алгоритм, математична модель.

**This article is about the determination of the process of deleting formulas of algorithms. The algorithm of computer deleting of formulas of algorithms was given. Synthesized, minimizing the mathematical model was synthesized, minimized and the algorithm of deleting of formulas of abstract algorithms was studied.**

**Key words:** deleting, algorithms, mathematical model.

### Вступ. Постановка проблеми

Відома [1,2] алгебра алгоритмів, 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 алгоритмів розроблено спеціалізовану комп'ютерну підсистему МОДАЛІ [3], але нею не знищуються формули абстрактних алгоритмів.

У таких графічних пакетах, як Microsoft Visio, Corel DRAW, Adobe Illustrator [5–7] не реалізовано системи знищення формул алгоритмів. Без реалізації таких функцій процеси редагування та набору формул алгоритмів значно ускладнюються.

### **Аналіз останніх досліджень та публікацій**

У роботах [8–10] описано підходи до розв’язання описаних задач, проте при всій різноманітності у жодному з наведених досліджень автори не проаналізували та не описали основних задач, що виникають при спробі адаптувати формули алгоритмів та знищувати їх.

При всій актуальності задачі на сьогоднішній день накопичено відносно невеликий досвід її розв’язання, який насамперед визначається відносно новим напрямком досліджень.

### **Формулювання цілей статті**

При спробі набору, редагування та знищення формул алгоритмів засобами відомих інформаційних систем виникають значні труднощі. Ціллю дослідження є синтез, дослідження та побудова математичного забезпечення процесу знищення формул алгоритмів.

### **Ілюстрація знищення формул абстрактних алгоритмів**

Процес знищення формул абстрактних алгоритмів є складовою процесу набору формул. За необхідності знищення формули алгоритму її необхідно вибрати і задати режим знищення (з мінімізацією чи без мінімізації).

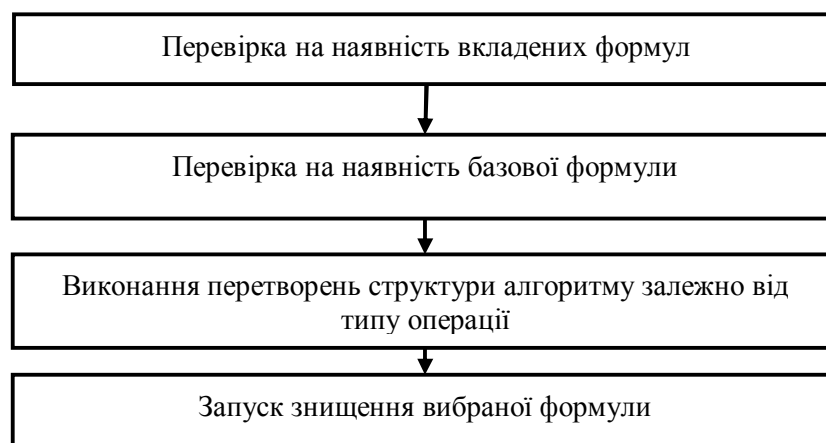
У редакторі формул алгоритмів мають бути реалізовані такі інструментальні засоби:

- вибору формули алгоритму;
- завдання режиму знищення формул алгоритмів.

Процес знищення формул алгоритмів у редакторі вимагає виконання таких дій:

- вибрати інструмент ідентифікації формули алгоритму чи унітерму;
- вибрати курсором миші необхідний абстрактний алгоритм;
- вибрати в випадному меню системи інструмент “Знищити”
- задати режим знищення (з мінімізацією чи без мінімізації)
- виконати запуск системи знищення формул абстрактних алгоритмів.

Схему процесу знищення формули абстрактного алгоритму наведено на рис. 1.



*Рис. 1. Схеми процесу знищення формули алгоритму*

На рис. 1 перший блок описує перевірку на наявність вкладених формул; другий блок описує перевірку на наявність базової формули; в третьому блоці описуються виконання перетворень структури алгоритму залежно від типу операції; в четвертому блоці описується запуск знищення вибраної формули.

Абстрактний алгоритм процесу знищення формул описує такі унітерми:

Унітерм  $P_1(i, c)$  присвоєння знищуваній формулі константи. Секвенція ( $S_1$ ), якою описується знищення формули в випадку базового знака елімінування або циклічних операцій, має вигляд:

$$S_1 = \left( \begin{array}{c} P_1(t_i, c_1) \\ ; \\ P_1(i, *) \end{array} \right),$$

де  $P_1(t_i, c_1)$  – унітерм присвоєння змінній типу вибраної формули ( $t_i$ ) константи ( $c_1$ ), а  $P_1(i, *)$  – унітерм присвоєння знищуваній формулі порожнього унітерму (\*);

Унітерми  $P_1(F_f^b, i)$  присвоєння другій вкладеній формулі ( $F_f^b$ ) в базову формулу другого рівня порядкового номера знищеної формули ( $i$ ) і  $P_1(F_f^a, i)$  присвоєння першій вкладеній формулі ( $F_f^a$ ) в базову формулу другого рівня порядкового номера знищеної формули елімінуються ( $L_1$ ) за умови перевірки, чи порядковий номер базової формули знищеного унітерму дорівнює порядковому номеру другої вложеної формули в базову формулу другого рівня ( $u_1$ ):

$$L_1 = \left| \begin{array}{c} P_1(F_f^b, i) ; P_1(F_f^a, i) ; u_1 - ? \end{array} \right|$$

Унітерми  $P_1(F_i^a, F_f)$  присвоєння першій вкладеній формулі ( $F_i^a$ ) в базову формулу знищеного унітерму порядкового номера базової формули другого рівня ( $F_f$ ) та  $P_1(F_i^b, F_f)$  присвоєння другій вкладеній формулі ( $F_i^b$ ) в базову формулу знищеного унітерму порядкового номера базової формули другого рівня елімінуються ( $L_2$ ) за умови перевірки, чи порядковий номер знищеного унітерму дорівнює порядковому номеру другої вкладеної формули в базову ( $u_2$ ):

$$L_2 = \left| \begin{array}{c} P_1(F_i^a, F_f) ; P_1(F_i^b, F_f) ; u_2 - ? \end{array} \right|$$

Унітерми  $P_1(F_i^a, c)$  присвоєння першій вкладеній формулі ( $F_i^a$ ) в базову формулу знищеного унітерму константи ( $c$ ) і  $P_1(F_i^b, i)$  присвоєння другій вкладеній формулі ( $F_i^b$ ) в базову формулу другого рівня константи елімінуються ( $L_3$ ) за умови перевірки, чи порядковий номер базової формули знищеного унітерму дорівнює порядковому номеру другої вкладеної формули в базову формулу другого рівня ( $u_3$ ):

$$L_3 = \left| \begin{array}{c} P_1(F_i^a, c) ; P_1(F_i^b, c) ; u_3 - ? \end{array} \right|$$

Секвенція, якою описується знищення формули у випадку, коли знищувана формула є вкладеною, має такий вигляд:

$$S_2 = \overbrace{L_2 ; L_1}.$$

Елімінування  $L_3$  і секвенцію  $S_2$  елімінуємо за умови перевірки, чи знищувана формула є вкладеною формулою ( $u_6$ ), у результаті чого отримаємо формулу:

$$L_4 = \left| \begin{array}{c} L_3 ; S_2 ; u_6 - ? \end{array} \right|$$

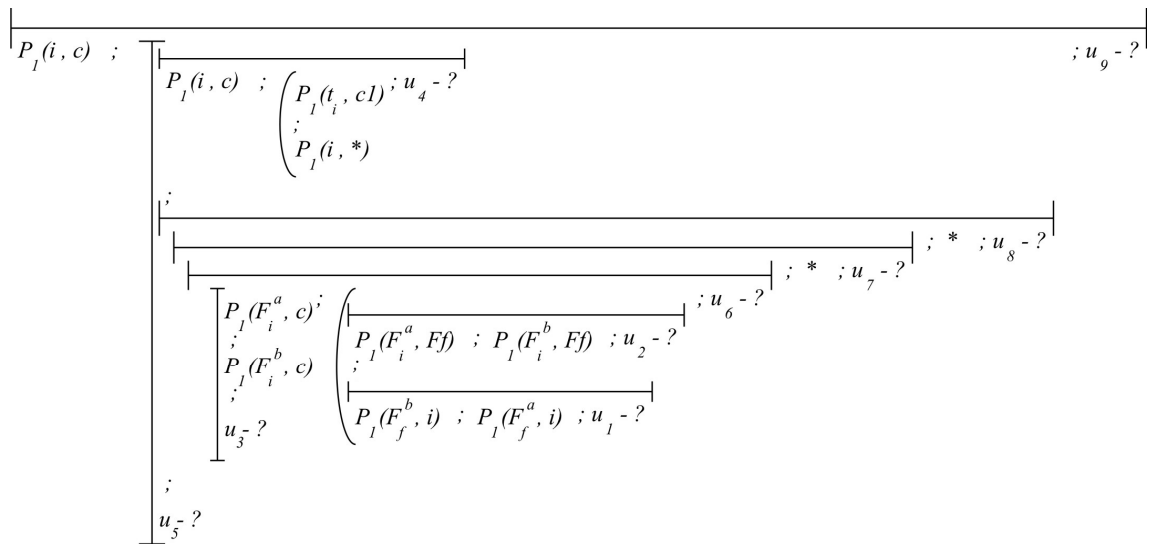
Елімінування  $L_4$  і порожній унітерм елімінуємо за умови перевірки, чи знищувана формула є унітермом ( $u_7$ ), і отримаємо формулу:

$$L_5 = \left| \begin{array}{c} L_4 ; * ; u_7 - ? \end{array} \right|$$

За умови перевірки на наявність базової формули в знищуваній формулі ( $u_8$ ) елімінуємо елімінування  $L_5$  і порожній унітерм:

$$L_6 = \left| \begin{array}{c} L_5 ; * ; u_8 - ? \end{array} \right|$$

Після підстановки відповідних елімінувань і на основі властивості дистрибутивності операції елімінування мінімізації отримаємо формулу абстрактного алгоритму, якою описується процес знищення формул алгоритмів. Синтезований абстрактний алгоритм зображено нижче.

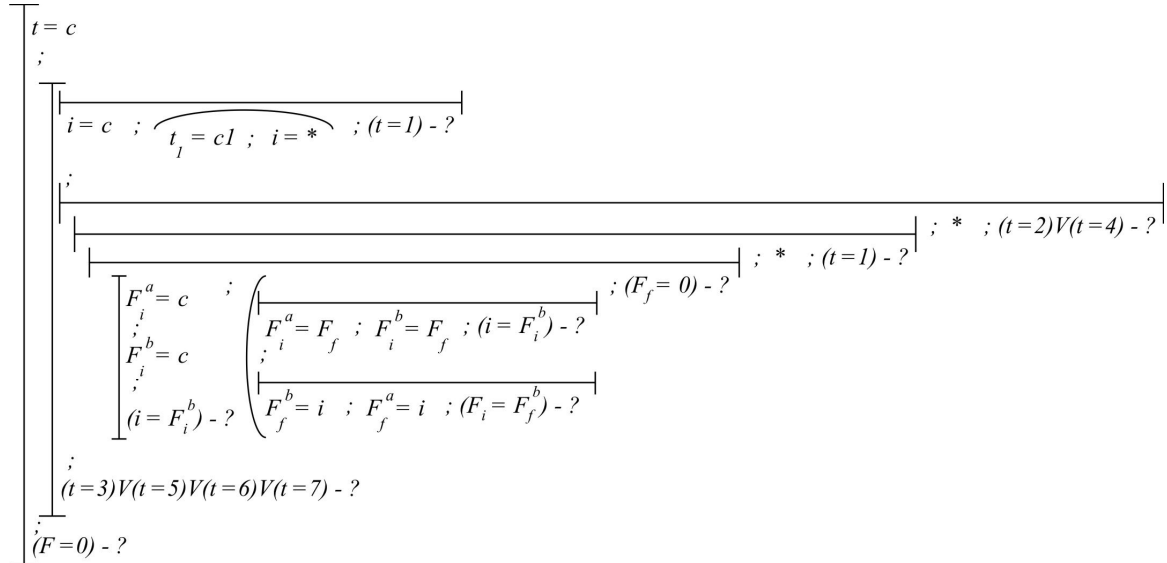


Після аналізу процесів редагування формул абстрактних алгоритмів у відомих системах набору та редагування та в спеціалізованій комп'ютерній системі МОДАЛ було зазначено, що процес знищення формул абстрактних алгоритмів в них не реалізується. Тому постала задача розроблення та реалізації моделі процесу знищення формул абстрактних алгоритмів.

Процес знищення формул алгоритмів користувач повинен викликати натисканням клавіші Delete на клавіатурі чи використовуючи відповідні команди випадного меню. Суть цього процесу полягає в перетворенні структури алгоритму (зв'язків між вложенними та базовими формулами) залежно від положення знищеного унітерму.

Побудуємо та дослідимо модель процесу знищення формул абстрактних алгоритмів.

Замінивши в абстрактному алгоритмі процесу знищення формул абстрактний унітерм предметним унітермом присвоєння ( $P_1(i, c)$  на  $i = c$ ,  $P_1(i, c_1)$  на  $i = c_1$  і т. д.), абстрактний умовний унітерм  $u_1$  предметним унітермом порівняння ( $F_i = F_f^b$ ) - ?, абстрактні умовні унітерм  $u_2$  і  $u_3$  предметним унітермом перевірки на наявність нижніх індексів ( $i = F_i^b$ ) - ?, унітерми  $u_4$  і  $u_7$  предметним унітермом перевірки на тип унітерму ( $t = 1$ ) - ?, унітерм  $u_5$  предметним унітермом перевірки на тип унітерму ( $t = 3$ ) $V(t = 5)$  $V(t = 6)$  $V(t = 7)$  - ?, унітерм  $u_6$  предметним унітермом перевірки на наявність базової формули другого рівня ( $Ff = 0$ ) - ?, унітерм  $u_8$  предметним унітермом перевірки на тип унітерму ( $t = 2$ ) $V(t = 4)$  - ?, унітерм  $u_9$  предметним унітермом перевірки на наявність базової формули ( $F = 0$ ) - ? і задавши секвентні області значень змінним, отримаємо модель абстрактного алгоритму процесу знищення унітермів.



Секвентні області значень змінних моделі процесу знищення унітермів матимуть вигляд:

$$i = F = F_i = F_f = F_i^a = F_i^b = F_f^a = F_f^b \in Q_1 = \overbrace{1 ; 2 ; \dots ; n}$$

де  $n = 1000$  – максимальне число унітермів в алгоритмі.

$$t \in Q_2 = \overbrace{1 , 2.. , 7} , c = 0, c_1 = 1$$

Дослідження опису моделлю процесу знищення унітермів виконано з використанням методу трансфінитної математичної індукції.

Пдана модель має такі змінні:  $i, F_f, F_i, F_i^a, F_i^b, F_f^a, F_f^b, F, t$ . Як видно з формули,  $F_i^b, F_f^a, F_f^b$  обчислюють за значеннями змінних  $i, F_f, t$  і констант  $c$  і  $c_1$ . Тому дослідження має бути виконане за цими змінними.

Нехай  $i = w < p \in Q_1, F_f = z < q \in Q_1, t = s < r \in Q_2, c = 0, c_1 = 1$ , а формула описує процес знищення унітермів.

Встановимо, чи моделлю описується процес знищення унітермів у випадку, коли базова формула є знаком операції секвентування ( $t = 4$ ) і в знищуваній формулі є базова формула другого рівня при значенні змінної  $F_f = z + 1 = q$ . Отримаємо вираз:

$$\left( \begin{array}{l} \overline{F_w^a = z + 1 ; F_w^b = z + 1 ; (w = F_w^b) - ?} \\ ; \\ \overline{F_f^b = w ; F_f^a = w ; (F_w = F_f^b) - ?} \end{array} \right) ,$$

як видно з формули, при зміні  $F_f$  на 1, значення порядкового номера другої вкладеної формули в базову також змінюється на цю величину залежно від вибраної для знищення першої чи другої вкладеної формули.

Встановимо, чи моделлю описується процес знищення унітермів у випадку, коли базова формула є знаком операції секвентування або паралелення ( $t = 2$ ) або ( $t = 4$ ) і в знищуваній формулі немає базової формули другого рівня при  $c = 0$ . Отримаємо вираз:

$$\overline{F_w^a = 0 ; F_w^b = 0 ; (w = F_w^b) - ?}$$

На підставі трансфінитної математичної індукції стверджуємо, що формула описує процес знищення унітермів для всіх можливих значень змінних.

Нехай дано таку формулу абстрактного алгоритму:

$$\left( \begin{array}{l} A \\ , \\ B \\ , \\ \overline{C ; D ; u - ?} \end{array} \right)$$

Спочатку дослідимо процес копіювання та вставляння модуля алгоритму.

- Ідентифікуємо знак операції секвентування;
- Виберемо команду “Копіювати”;
- Кліком лівої клавіші миші виберемо майбутнє місце формули;
- Виберемо команду “Вставити”.

У результаті отримаємо таку формулу:

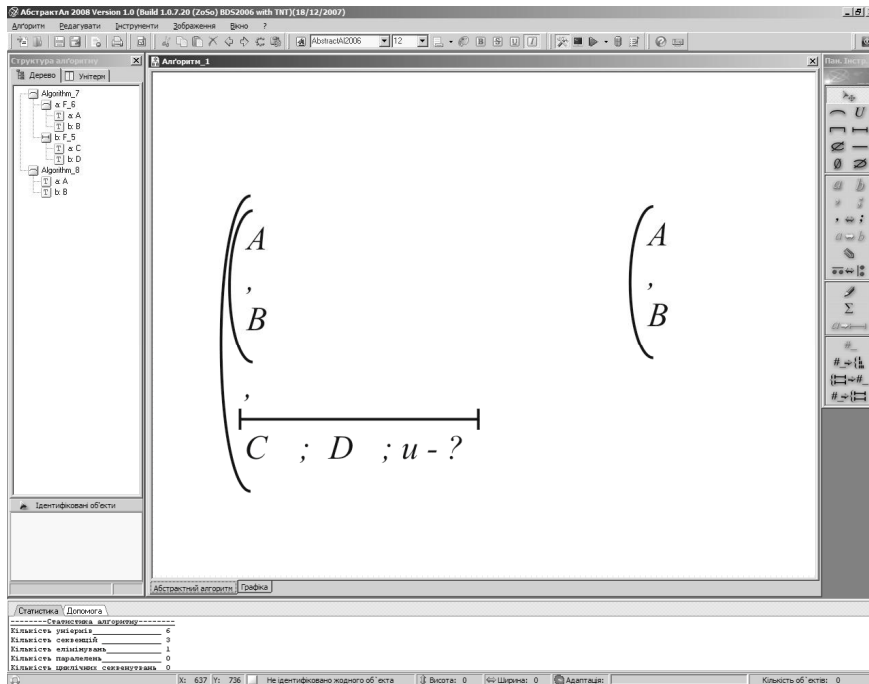


Рис. 2. Вставка з буфера обміну формули абстрактного алгоритму

Тепер дослідимо процес вирізання та знищення формули абстрактного алгоритму:

- Ідентифікуємо знак операції секвенування;
- Виберемо команду “Вирізати”;
- Кліком лівої клавіші миші виберемо майбутнє місце формули;
- Виберемо команду “Вставити”.

Результатом вищеприказаних дій буде така формула абстрактного алгоритму:

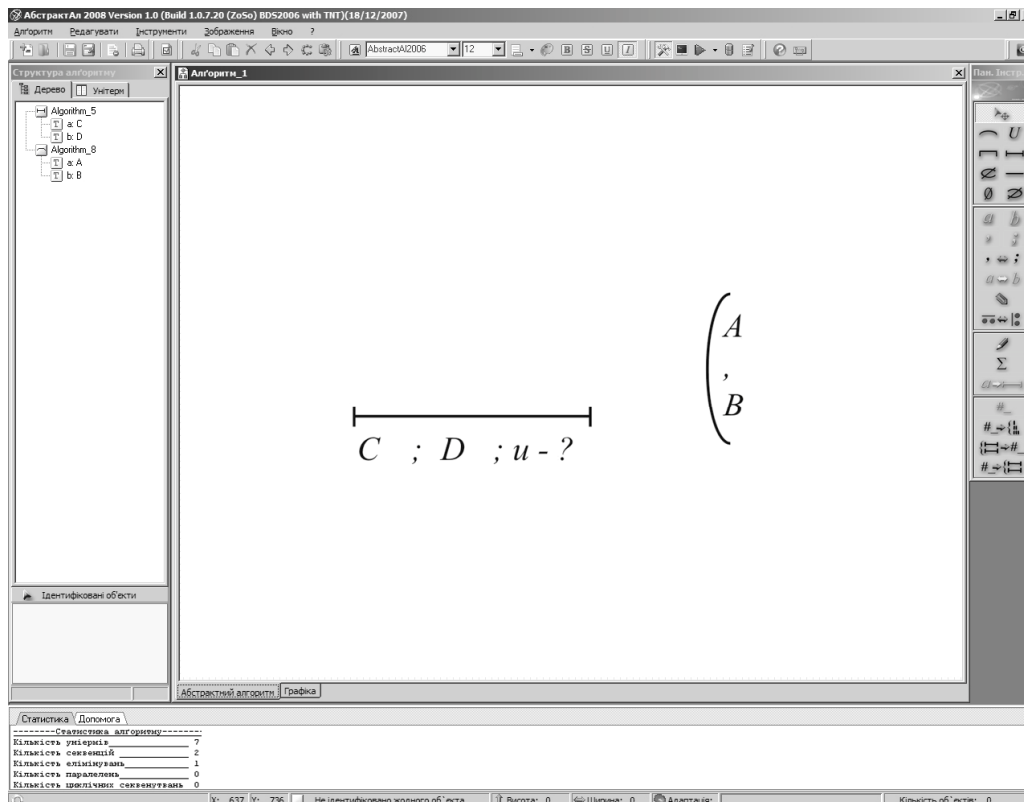


Рис. 3. Вирізання та вставка формули абстрактного алгоритму

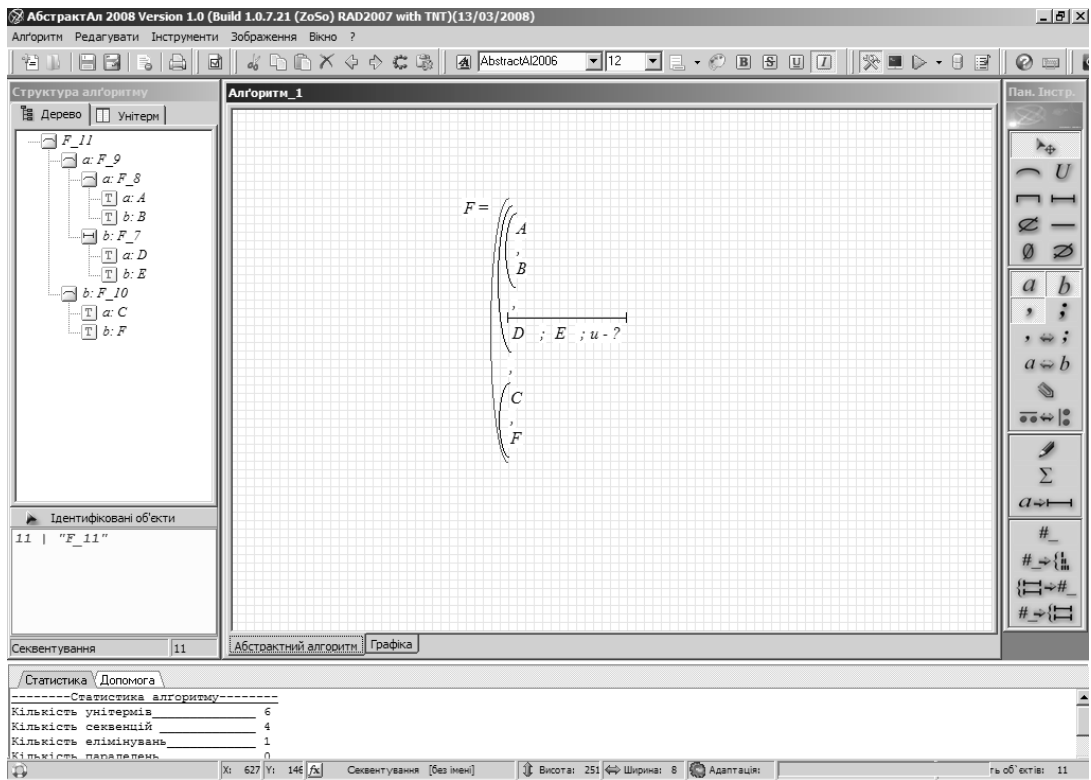


Рис. 4. Вигляд формули до знищення

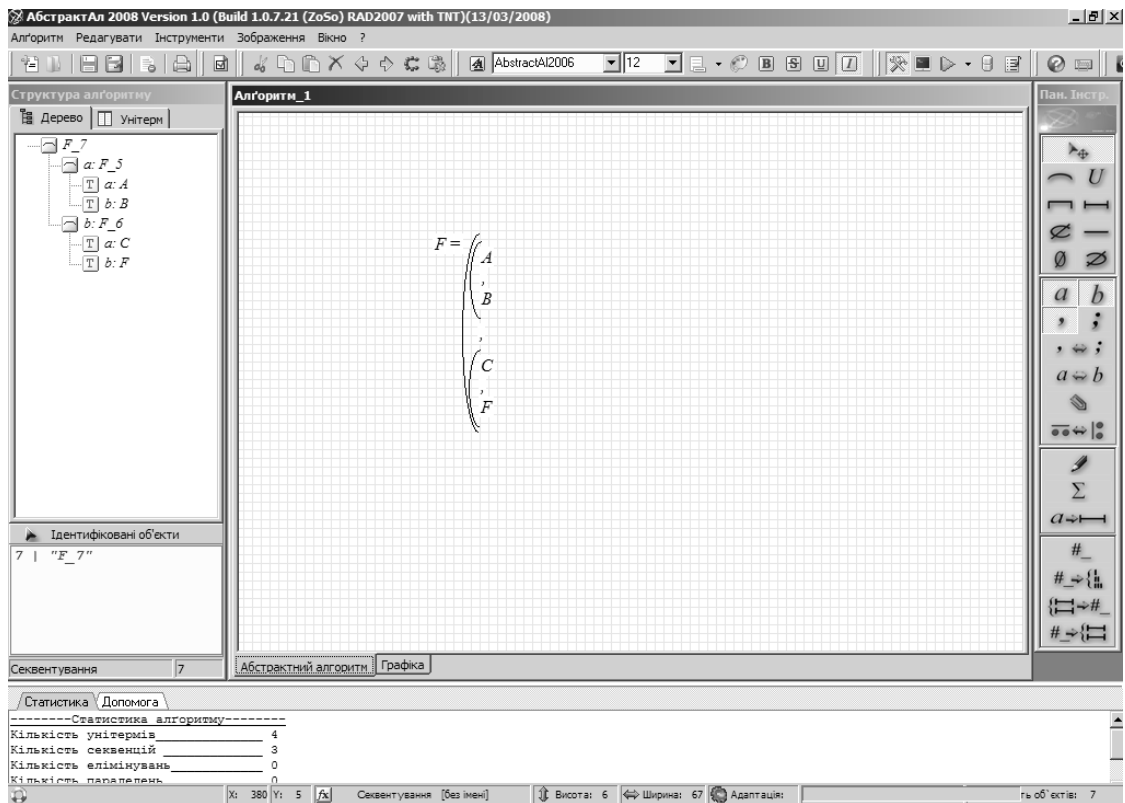


Рис. 5. Вигляд формули після знищення елімінування

## Висновки

1. Синтезована, мінімізована і досліджена математична модель алгоритму знищення формул абстрактних алгоритмів є складовою процесів редагування формул абстрактних алгоритмів

(вирізання, копіювання, вставляння і знищення), реалізація якої забезпечує більшу наочність подання алгоритмів у вигляді формул теорії абстрактних алгоритмів.

2. Дослідження математичної моделі ще до її практичної реалізації і апробації забезпечило виявлення помилок, допущених у процесі її синтезу, та доводить, що вона описує необхідні процеси.

3. Абстрактним алгоритмом знищення формул абстрактних алгоритмів описано процеси знищення формул алгоритмів.

1. Овсяк В.К. Синтез і дослідження алгоритмів комп'ютерних систем / В.К. Овсяк, В.М. Бритковський, О.В. Овсяк, Ю.В. Овсяк.. – Львів, 2004. – 276 с. 2. Овсяк В.К. АЛГОРИТМИ: методи побудови, оптимізації, дослідження вірогідності/Овсяк В.К. – Львів: Світ, 2001. – 160 с. 3. Бритковський В. М. Моделювання редактора формул секвенційних алгоритмів : Автореф. дис. на здобуття наук. ступеня канд.техн.наук : спец. 01.05.02 “Математичне моделювання та обчислювальні методи” – Львів: видавничо-поліграфічний відділ ЛьЦНТЕІ. – 18 с., 4.Солоницын Ю.А. Microsoft Visio 2007. Создание деловой графики / Солоницын Ю.А. – СПб.:Питер, 2009. – 160 с. 5.Миронов Д.Ф. CorelDRAW X3.Учебный курс / Миронов Д.Ф.-СПб.:Питер, 2006. – 397 с, 6.Жвалевский А.В. Adobe Illustrator CS в теории и на практике/ А.В.Жвалевский, Ю.А. Гурский, Г.Б. Корабельникова.-М.:Новое знание, 2004.-607с. 7. А.Василюк, Т.Басюк Підсистема адаптації формул алгоритмів / Василюк А., Басюк Т. // Інформаційні системи та мережі. – Львів: Нац. ун-т “Львівська політехніка”, 2010.– № 689. – С. 36–43 8. Василюк А., Басюк Т. Алгоритми згортання і розгортання формул абстрактних алгоритмів / А. Василюк, Т. Басюк // Інформаційні системи та мережі. – Львів: Нац. ун-т “Львівська політехніка”, 2009.– № 653. – С. 37–45. 9. Василюк А. Абстрактний алгоритм редактора формул абстрактних алгоритмів “АбстрактАл” / А. Василюк // Комп'ютерні технології друкарства : Збірник наукових праць. – Львів: УАД, 2006. – № 16. – С. 99–108 .10. Овсяк, А. Василюк // Комп'ютерні технології друкарства : Збірник наукових праць.– Львів: УАД, 2004. – №12. – С. 137–145.