

КОМП'ЮТЕРНА ТА МАТЕМАТИЧНА ЛІНГВІСТИКА

УДК 004.942

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

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ПРОЦЕСУ ТРАНСФОРМУВАННЯ ФОРМУЛ АЛГЕБРИ АЛГОРИТМІВ

© Василюк А. С., Басюк Т. М., 2018

Описано процеси трансформування формул алгебри алгоритмів. Наведено алгоритм обчислення геометричних параметрів унітермів. Синтезовано, мінімізовано, побудовано математичну модель і досліджено алгоритм процесу трансформування формул алгебри алгоритмів.

Ключові слова: унітерм, алгоритм, математична модель.

The processes of transforming the algorithm algebra formulas are described. The algorithm for calculation of geometrical parameters of unithrams is given. Synthesized, minimized, built a mathematical model and investigated the algorithm of the transformation process of algebra algorithm formulas.

Key words: unitherm, algorithms, mathematical model.

Вступ. Загальна постановка проблеми

Проблема редагування та створення формул алгебри алгоритмів за відсутності спеціалізованих засобів є актуальною. Відомі графічні системи не мають засобів, що реалізують процеси трансформування формул алгебри алгоритмів зі збереженням їхньої структури. Такі можливості дозволять у будь-який час повернути формулу алгоритмів до початкового стану на будь-якому етапі роботи над нею.

Зв'язок висвітленої проблеми

із важливими науковими та практичними завданнями

Враховуючи описані вище недоліки, з метою зменшення затрат було розроблено спеціалізовану комп'ютерну підсистему АБСТРАКТАЛІ [4,6,7]. Використовуючи цю підсистему, процеси набору та редагування формул абстрактних алгоритмів значно скорочені за часом та за кількістю операцій, необхідних для здійснення цих процесів. Однак засоби цієї підсистеми не забезпечують трансформування формул абстрактних алгоритмів. Тому темою цієї статті є моделювання процесу трансформування формул алгебри алгоритмів.

З огляду на це, описана науково-практична задача є трудомістким та неперервним процесом. Розв'язання поставленої задачі дасть змогу сформулювати методологічне підґрунтя для створення систем редагування та створення редакторів формул алгоритмів з можливістю роботи з великою кількістю формул.

Аналіз останніх досліджень та публікацій

Алгебра алгоритмів [1,2,3] має специфічні знаки операцій, яких не існує серед відомих позначень математичних операцій. Наприклад, знаки операцій секвентування, елімінування, паралелення та циклічні операції [1,2,3]. Засоби цієї теорії дають можливість подати алгоритм у вигляді формул. Для набору та редагування таких формул алгоритмів можна використовувати засоби універсальних систем, але такий набір є трудомістким і потребує багато затрат як робочого, так і комп'ютерного часу.

Основні завдання дослідження та їх значення

Мета дослідження – визначити основні способи трансформування формул алгебри алгоритмів. Проведене дослідження надасть засоби для створення систем із можливостями роботи з великими формулами, які в перспективі можна оптимізувати за розміром.

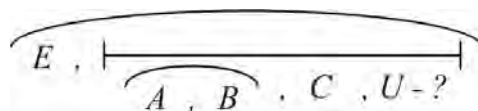
Основні результати досліджень

Формули абстрактних алгоритмів складаються з *елементів* [5]. Елементи можуть бути *вложеними* і *базовими* [5]. Крім цього, елементи мають свої *параметри* [5]. Процес трансформування формул абстрактних алгоритмів полягає в зміні структури формули абстрактного алгоритму залежно від дій, проведених над алгоритмом. Нижче проілюстровано декілька прикладів процесів трансформувань формул абстрактних алгоритмів.

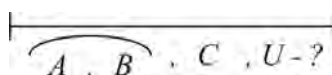
Нехай дано таку формулу абстрактного алгоритму:



У випадку видалення одного з термів, наприклад, D , формула абстрактного алгоритму має мати такий вигляд:



Коли видалити терм E , базова секвенція повинна спроститися, тобто:



У випадку, коли формула абстрактного алгоритму складається лише зі знаку операції секвентування та двох вложених термів, після видалення одного з них, має залишитись тільки другий терм, а знак операції секвентування спроститися.

Структура алгоритму процесу трансформування формул абстрактних алгоритмів

Нижче зображено структуру алгоритму процесу трансформування формул абстрактних алгоритмів. У структурі алгоритму описано процес трансформування формули залежно від типу операції базової формули абстрактного алгоритму, від наявності базової формули і її типу операції. Кожен із цих параметрів описано в структурі даних [5].

Синтез алгоритму процесу трансформування формул абстрактних алгоритмів

Синтез абстрактних алгоритмів виконують двома методами [3]: секвенційним і табличним. Секвенційний метод простіший, отже, вибираємо саме його. Він передбачає виконання синтезу в два етапи: синтез секвенцій і синтез елімінувань.

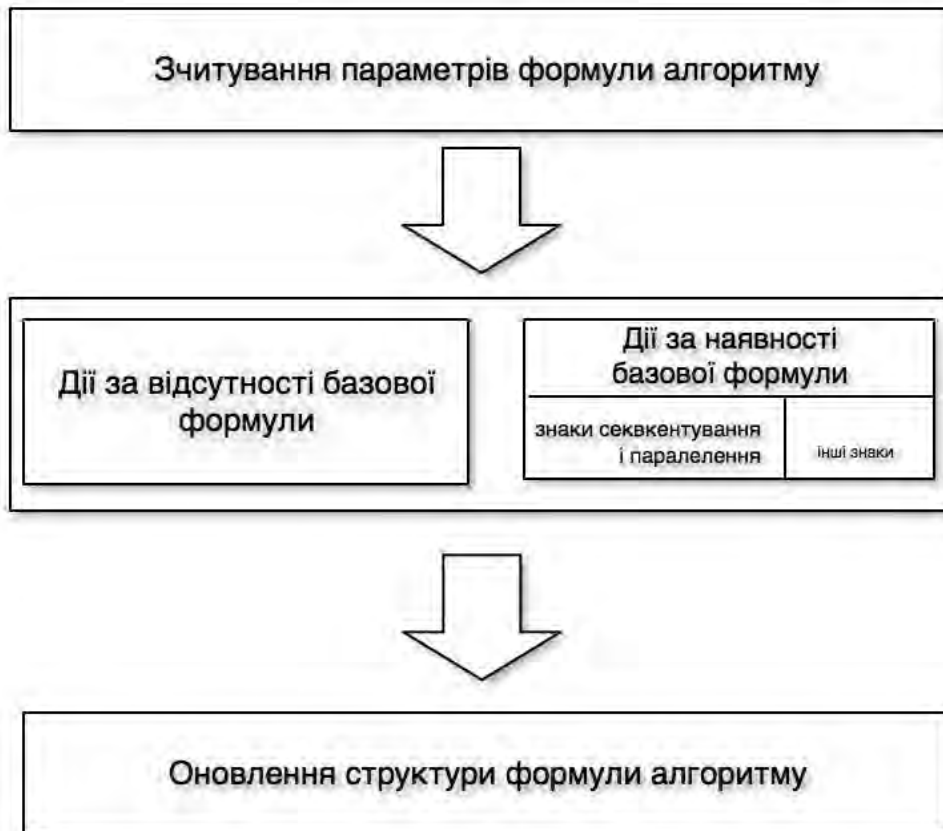


Рис. 1. Структура алгоритму процесу трансформування формул абстрактних алгоритмів

Синтез секвенцій. Абстрактний алгоритм процесу трансформування формул абстрактних алгоритмів включає такі секвенції:

S_1 – секвенція трансформування формул абстрактних алгоритмів у випадку, коли вибрана формула є другим вложеним унітермом,

$$\left(\begin{array}{l} App(T, A(P(Ac))) \\ \vdots \\ App(P(A(P(Ac))), P(P(Ac))) \end{array} \right);$$

S_2 – секвенція трансформування формул абстрактних алгоритмів у випадку, коли вибрана формула є першим вложеним унітермом,

$$\left(\begin{array}{l} App(T, B(P(Ac))) \\ \vdots \\ App(P(B(P(Ac))), P(P(Ac))) \end{array} \right);$$

S_3 – послідовність необхідних дії для видалення зайвих операцій.

$$\overbrace{D(P(Ac)) \ ; \ D(Ac)}$$

S_4 – секвенція трансформування формул абстрактних алгоритмів у випадку, коли вибрана формула не є термом

$$\left(\begin{array}{l} App(To(Ac), 1) \\ ; \\ App(Text(Ac), *) \\ ; \\ App(V(Ac), 0) \end{array} \right)$$

Синтез елімінувань. Секвенції S_1 і S_2 елімінуються елімінуванням L_1 за умовою перевірки на вибір другої вложеної формули $U_1(Ac, B(P(Ac))) - ?$ Секвенція S_4 і терм $App(Text(Ac), *)$ елімінуються елімінуванням L_2 за умовою перевірки на тип операції вибраної формули $U_1(To(Ac), 1) - ?$ Запишемо ці елімінування:

$$L_1 = \left| \begin{array}{l} \left(\begin{array}{l} App(T, A(P(Ac))) \\ ; \\ App(P(A(P(Ac))), P(P(Ac))) \end{array} \right) , \left(\begin{array}{l} App(T, B(P(Ac))) \\ ; \\ App(P(B(P(Ac))), P(P(Ac))) \end{array} \right) , U_1(Ac, B(P(Ac))) - ? \end{array} \right|$$

$$L_2 = \left| \begin{array}{l} App(Text(Ac), *) , \left(\begin{array}{l} App(To(Ac), 1) \\ ; \\ App(Text(Ac), *) \\ ; \\ App(V(Ac), 0) \end{array} \right) , U_1(To(Ac), 1) - ? \end{array} \right|$$

Виконавши підстановку відповідних секвенцій в елімінування та мінімізацію за кількістю унітермів, отримаємо абстрактний алгоритм.

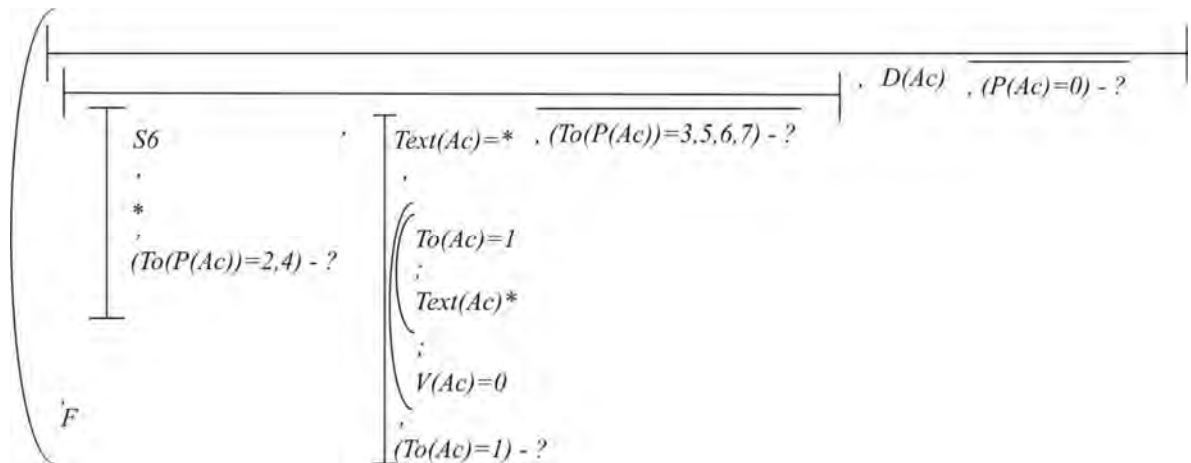
$$\left(\begin{array}{l} \left| \begin{array}{l} S5 \\ ; \\ * \\ ; \\ U_1(To(P(Ac)), \{2,4\}) - ? \end{array} \right| , \left| \begin{array}{l} App(Text(Ac), *) , U_1(To(P(Ac)), \{3,5,6,7\}) - ? \\ ; \\ \left(\begin{array}{l} App(To(Ac), 1) \\ ; \\ App(Text(Ac), *) \\ ; \\ App(V(Ac), 0) \end{array} \right) \\ ; \\ U_1(To(Ac), 1) - ? \end{array} \right| , D(Ac) , U_1(P(Ac), 0) - ? \end{array} \right)$$

$$\begin{array}{l}
 S5 = \left[\begin{array}{l}
 \left(\begin{array}{l}
 App(T, A(P(Ac))) \\
 \vdots \\
 App(P(A(P(Ac))), P(P(Ac)))
 \end{array} \right) ; \left[\begin{array}{l}
 App(B(P(P(Ac))), T) \\
 \vdots \\
 App(A(P(P(Ac))), T) \\
 \vdots \\
 U1(P(Ac), B(P(P(Ac)))) - ?
 \end{array} \right] , \left[\begin{array}{l}
 App(P(A(P(Ac))), 0) , U1(P(P(Ac)), 0) - ? \\
 \vdots \\
 App(P(B(P(Ac))), 0) \\
 \vdots \\
 U1(Ac, B(P(Ac))) - ?
 \end{array} \right] \\
 \left(\begin{array}{l}
 App(T, B(P(Ac))) \\
 \vdots \\
 App(P(B(P(Ac))), P(P(Ac)))
 \end{array} \right) \\
 \vdots \\
 U1(Ac, B(P(Ac))) - ? \\
 \left. \begin{array}{l}
 \left(D(P(Ac)) ; D(Ac) \right) \\
 \vdots \\
 * \\
 \vdots \\
 U1(To(Ac), 1) - ?
 \end{array} \right]
 \end{array} \right.
 \end{array}$$

Побудова моделі здійснюється заміною абстрактних термів на предметні. Тобто терми типу $Z(k, x)$ на $k = x$ та $U(p, 1) - ?$ на $(p = 1) - ?$.

У результаті дістанемо таку математичну модель:

$$\begin{array}{l}
 S6 = \left[\begin{array}{l}
 \left(\begin{array}{l}
 T = A(P(Ac)) \\
 \vdots \\
 P(A(P(Ac))) = P(P(Ac))
 \end{array} \right) ; \left[\begin{array}{l}
 B(P(P(Ac))) = T \\
 \vdots \\
 A(P(P(Ac))) = T \\
 \vdots \\
 (P(Ac) = B(P(P(Ac)))) - ?
 \end{array} \right] , \left[\begin{array}{l}
 P(A(P(Ac))) = 0 , (P(P(Ac))) = 0 - ? \\
 \vdots \\
 P(B(P(Ac))) = 0 \\
 \vdots \\
 (Ac = B(P(Ac))) - ?
 \end{array} \right] \\
 \left(\begin{array}{l}
 T = B(P(Ac)) \\
 \vdots \\
 P(B(P(Ac))) = P(P(Ac))
 \end{array} \right) \\
 \vdots \\
 (Ac = B(P(Ac))) - ? \\
 \left. \begin{array}{l}
 \left(D(P(Ac)) ; D(Ac) \right) \\
 \vdots \\
 * \\
 \vdots \\
 (To(Ac) = 1) - ?
 \end{array} \right]
 \end{array} \right.
 \end{array}$$



Висновок

У результаті проведеного дослідження синтезовано модель процесу трансформування формул абстрактних алгоритмів. Мінімізована модель описує процес трансформування формул абстрактних алгоритмів. Використанням синтезованої моделі процесу трансформування формул абстрактних алгоритмів забезпечуються менші затрати часу за ручного коректування формули після видалення терму та ефективніше використання середовища розробки [6, 7]. Актуальним є завдання подальшого переходу від автоматичного спрощення шляхом видалення термів у формулі абстрактного алгоритму до автоматичного спрощення шляхом оцінки логічного змісту формули абстрактного алгоритму. Абстрактним алгоритмом процесу трансформування формул абстрактних алгоритмів описано процес автоматичного трансформування шляхом видалення унітермів у формулах абстрактних алгоритмів.

1. Овсяк В. Синтез і дослідження алгоритмів комп'ютерних систем // В. Овсяк., О. Овсяк, В. Бритковський – Львів, 2004. – 276 с.
2. Овсяк В. АЛГОРИТМИ: методи побудови, оптимізації, дослідження вірогідності / В. Овсяк – Львів: Світ, 2001. – 160 с.
3. Овсяк В. Теорія секвенційних алгоритмів і проектування комп'ютерних систем / В. Овсяк, Ю. Овсяк, В. Бритковський. – Львів: УАД, 2001. – 142 с.
4. Бритковський В. М. Моделювання редактора формул секвенційних алгоритмів: автореф. дис. роб. к. т. н. – Львів: видавничо-поліграфічний відділ ЛвЦНТЕІ. – 18 с.
5. Василюк А. С. Принцип побудови підсистеми редагування формул абстрактних алгоритмів / А. Василюк, В. Овсяк // Комп'ютерні технології друкарства. – Львів: УАД, 2004. – № 12. – С. 137–146.
6. Василюк А. С. Інтелектуальний аналіз структури даних та математичного забезпечення редактора формул алгоритмів / А. С. Василюк. – Львів: Нац. ун-т “Львівська політехніка”, 2015. – № 832. – С. 34–48
7. Basyuk T. M. Graph visualization in a heterogeneous environment by means of algebra algorithms // T. M. Basyuk, A. S. Vasylyuk // Science and Education a New Dimension. Natural and Technical Sciences. – Hungary: Budapest – IV(10), Issue: 91, 2016. – 66–71 pp.