

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ СТРУКТУРИ ДАНИХ ТА МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ РЕДАКТОРА ФОРМУЛ АЛГОРИТМІВ

© Василюк А. С., 2015

Описано означення процесу адаптації, структури даних редактора алгоритмів. Синтезовано, мінімізовано, побудовано математичну модель структури даних редактора формул алгоритмів, ієрархічно-модульний принцип побудови редактора, процесу адаптації, згортання і розгортання формул алгебри алгоритмів.

Ключові слова: унітерм, алгоритм, математична модель.

This article is about determination of process of adaptation, data structure of algorithms editor. Mathematical model of data structure of algorithms editor, hierarchical-modular construction principle of editor, of adaptation process and expanding and collapsing processes were synthesized, minimized and built.

Key words: unitherm, algorithms, mathematical model.

Вступ. Загальна постановка проблеми

Алгебра алгоритмів [1,2], об'єктами якої є алгоритми, описані як формули, які можна перетворити, наприклад, з метою мінімізації, а також під час заміни, згортання і розгортання формул алгоритмів. Ця теорія має конкретні ознаки таких операцій, як секвентування, елімінування, паралелення та циклічні операції, які відображаються у вигляді спеціальних знаків, яких немає серед відомих математичних символів. З метою спрощення процесів набору і редагування формул алгоритмів необхідно створити структуру даних та математичне забезпечення редактора формул алгебри алгоритмів.

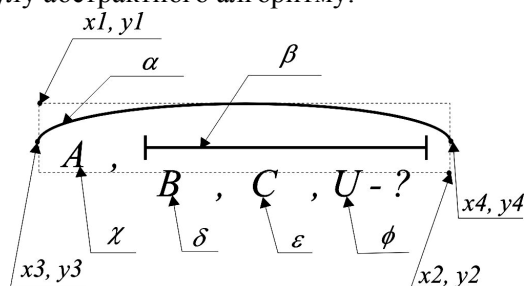
Аналіз останніх досліджень та публікацій

У роботах [3–6] описано підходи до розв'язання описаних задач, проте не виконано повного аналізу структури даних і математичного забезпечення редактора формул алгебри алгоритмів.

Незважаючи на актуальність задачі, сьогодні накопичено відносно невеликий досвід її розв'язання, який насамперед визначається відносно новим напрямком досліджень.

Формулювання цілей статті

Розглянемо таку формулу абстрактного алгоритму:



де $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$ – координати знаку операції секвентування; a – знак операції секвентування; b – знак операції елімінування; c, d, e, f – тривіальні унітерми [1, 2].

Своєю чергою, унітерми можуть бути нетривіальними. Цей випадок зображено нижче:

$$F_1(x_1 y^2).$$

У цьому випадку нетривіальний унітерм поділяється на такі складові унітерму: F , x , y – тривіальні унітерми, $1, 1, 2$ – індекси. Кожна із складових виразу має свої розміри: висоту та ширину. Всі ці параметри описуються модифікованою структурою даних, яку подано у вигляді табл. 1.

Таблиця 1

Структура даних редактора формул алгебри алгоритмів

Елемент структури даних	Опис
<i>id</i>	Унікальний ідентифікатор унітерму
<i>TypeObj</i>	Тип унітерму
<i>Inv, UInv, UElim</i>	Інвертована операція, умова й елімінування
<i>vertical</i>	Орієнтація знаку операції
<i>id_a, id_b</i>	Унікальні ідентифікатори вкладених унітермів
<i>IsOp</i>	Унікальний ідентифікатор батьківського унітерму
<i>x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12</i> <i>y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12</i>	Геометричні координати унітерму
<i>x_index_r_down1, y_index_r_down1;</i> <i>x_index_r_up1, y_index_r_up1;</i> <i>x_index_r_down2, y_index_r_down2;</i> <i>x_index_r_up2, y_index_r_up2;</i> <i>x_index_r_down3, y_index_r_down3;</i> <i>x_index_r_up3, y_index_r_up3;</i>	Геометричні координати індексів нетривіального унітерму
<i>H_t, W_t, H_i_up, W_i_up, H_i_down, W_i_down;</i> <i>H_t1, W_t1, H_t2, W_t2, H_t3, W_t3;</i> <i>H_i_up1, W_i_up1, H_i_down1, W_i_down1;</i> <i>H_i_up2, W_i_up2, H_i_down2, W_i_down2;</i> <i>H_i_up3, W_i_up3, H_i_down3, W_i_down3;</i>	Геометричні параметри (висота і ширина) індексів нетривіального унітерму
<i>Height, Width;</i> <i>Height1, Height2, Height3;</i> <i>Width1, Width2, Width3;</i>	Геометричні параметри всього унітерму і трьох складових нетривіального унітерму
<i>Color</i>	Колір унітерму
<i>text1,text2,text3</i>	Три складові нетривіального унітерму
<i>Separator</i>	Розділовий знак між унітермами
<i>Font</i>	Гарнітура шрифту унітерму
<i>F_Name</i>	Назва шрифту
<i>Umova</i>	Умовний унітерм для операції елімінування
<i>Zminna</i>	Унітерм змінної для циклічних операцій
<i>AfterU</i>	Наявність післяумови
<i>AfterUType</i>	Тип післяумови
<i>HeightCycl, WidthCycl</i>	Геометричні параметри знаку циклу
<i>SeqObl</i>	Секвентна область
<i>InputParameters</i>	Вхідні параметри
<i>OutputParameters</i>	Вихідні параметри

Використовуючи цю структуру даних, можна описати будь-який унітерм, враховуючи зв'язки між вкладеними і батьківськими унітермами.

У поле *TypeObj* заносять інформацію про тип об'єкта, а власне: 1 – ідентифікатор терма; 2 – операції паралелення; 3 – елімінування; 4 – секвентування; 5 – циклічного секвентування; 6 – циклічного елімінування; 7 – циклічного паралелення; 8 – інвертування.

За даними полів *id_a*, *id_b* і *IsOp* встановлюють зв'язки, які використовують для реалізації процесу адаптації формул алгебри алгоритмів. Тобто в поля *id_a*, *id_b* базового знаку операції алгоритму записують унікальні ідентифікаційні номери вкладених елементів, а в поле *IsOp* вкладеної формули записують унікальний ідентифікаційний номер батьківського унітерму.

Для цього прикладу поля, що відповідають за реалізацію зв'язків, в формулі алгебри алгоритмів виглядатимуть, як на рис. 1.

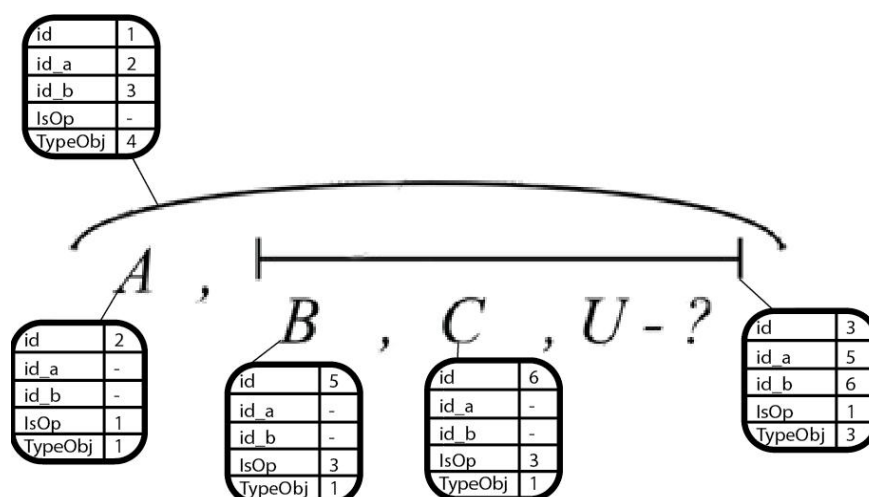
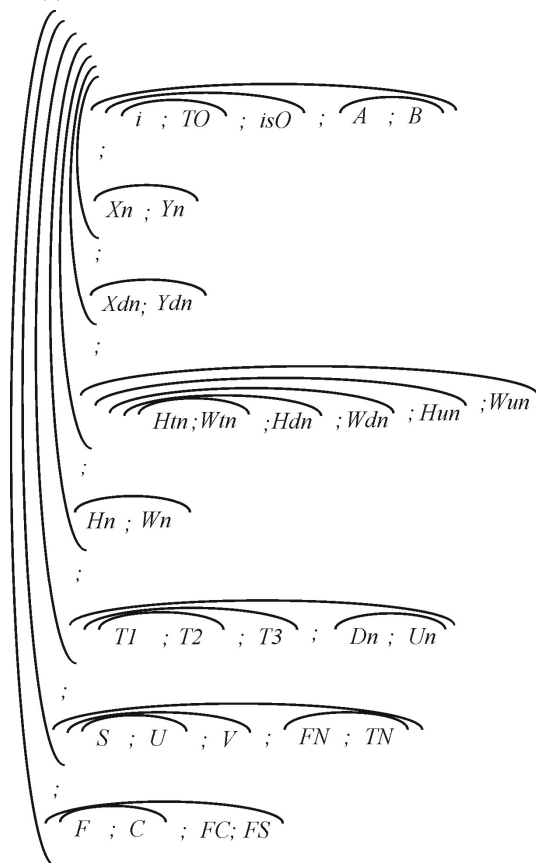


Рис. 1. Значення полів структури даних

Структура даних редактора формул абстрактних алгоритмів описує такі параметри формул алгоритмів чи унітермів:

- унікальний ідентифікатор формули (i);
- TO – тип формули (секвентування, елімінування, паралелення, циклічне секвентування, циклічне елімінування, циклічне паралелення та унітерм);
- ідентифікатор базової формули абстрактного алгоритму (isO);
- ідентифікатор першої вкладеної формули (A);
- ідентифікатор другої вкладеної формули (B);
- геометричні параметри (координати) об'єкта (залежно від типу об'єкта кількість координат змінюється), позначимо їх $Xn=x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12, y1, y2, y3, y4, y5, y6, y7, y8, y9, y10, y11, y12$;
- геометричні параметри верхніх та нижніх індексів унітермів ($Xdn=xd1, xd2, xd3, Ydn=yd1, yd2, yd3, Xun=xu1, xu2, xu3, Yun=yu1, yu2, yu3$);
- ширини та висоти окремо кожної складової унітерму (мається на увазі, що унітерм умовно поділяють на унітерми та на його індекси), ($Htn=Ht, Ht1, Ht2, Ht3, Wtn=Wt, Wt1, Wt2, Wt3, Hdn=Hd, Hd1, Hd2, Hd3, Wdn = Wd, Wd1, Wd2, Wd3, Hun = Hu, Hu1, Hu2, Hu3, Wun = Wu1, Wu2, Wu3$);
- сумарні ширини та висоти унітермів із врахуванням ширин і висот їх індексів ($Hn=H, H1, H2, H3, Wn=W, W1, W2, W3$);
- вміст трьох складових унітермів ($T1, T2, T3$);
- вміст індексів ($d1, d2, d3, u1, u2, u3$);
- розділовий знак між вкладеними формулами (S);
- унітерм умови (лише в знаках операцій елімінування, циклічного секвентування, циклічного елімінування та циклічного паралелення) (U);
- вміст унітерму змінної (лише для циклічних операцій) (V);
- унітерм назви формули (FN);
- унітерм ідентифікатора вітки дерева формули абстрактного алгоритму (TN);
- тип шрифту унітерму (F);
- колір знаку операції (C);
- колір шрифту унітерму (FC);
- тип накреслення шрифту (FS).

Структура даних редактора формул абстрактних алгоритмів, відображена засобами алгебри алгоритмів, матиме такий вигляд:



На основі структури даних створено ієрархічно-модульну модель редактора формул алгебри алгоритмів.

Модель тривірнева: перший рівень є загальносистемним, другий – функціональним і третій – термінальним.

Модель системи утворено такими модулями:

- функціональним модулем;
- модулем генерування унітерму;
- модулем додаткових операцій;
- системним модулем;
- модулем організаційних даних редактора.

Своєю чергою, ці модулі поділяють на:

- модуль ініціалізації редактора – модуль параметрів завантаження редактора;
- модуль допомоги користувачу – інструкції щодо користування редактором та основні положення алгебри алгоритмів;
- модуль ресурсних операцій – зберігання і зчитування параметрів знаків операцій;
- модуль налаштування робочого поля – модуль параметрів редактора, тобто колір канви робочого поля, шрифт системи та ін.
- модуль конвертування;
- функціональний модуль – основний модуль системи, який містить основні функціональні засоби редактора;
- модуль інформації – модуль, який містить системну інформацію та допоміжні засоби щодо роботи редактора;
- модуль генерування унітермів – модуль, засобами якого набирають та редагують унітерми.

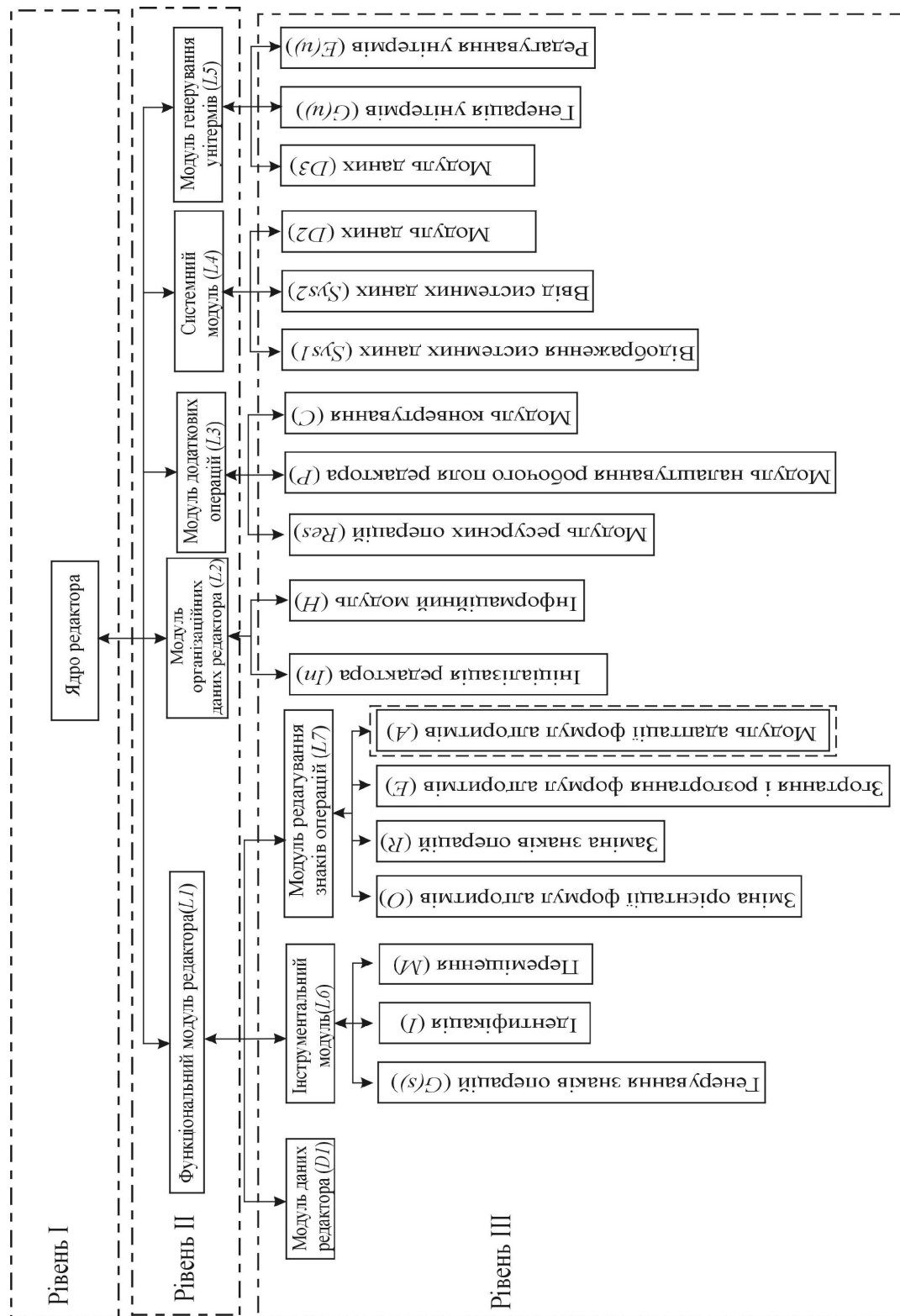


Рис. 2. Ієрархічно-модульна модель редактора

Функціональний модуль є основним модулем підсистеми редагування абстрактних алгоритмів. Він складається з таких модулів:

- модуль даних редактора – розділ редактора, в якому зберігається структура даних та інші параметри редактора;

- модуль генерації об'єктів, який, своєю чергою, поділяється на модулі генерування знаків операцій та модуль ідентифікації об'єктів;

- модуль редагування знаків операцій містить операції переміщення, зміну орієнтації формул алгоритмів, заміну знаків операцій, згортання і розгортання та модуль адаптації формул алгоритмів.

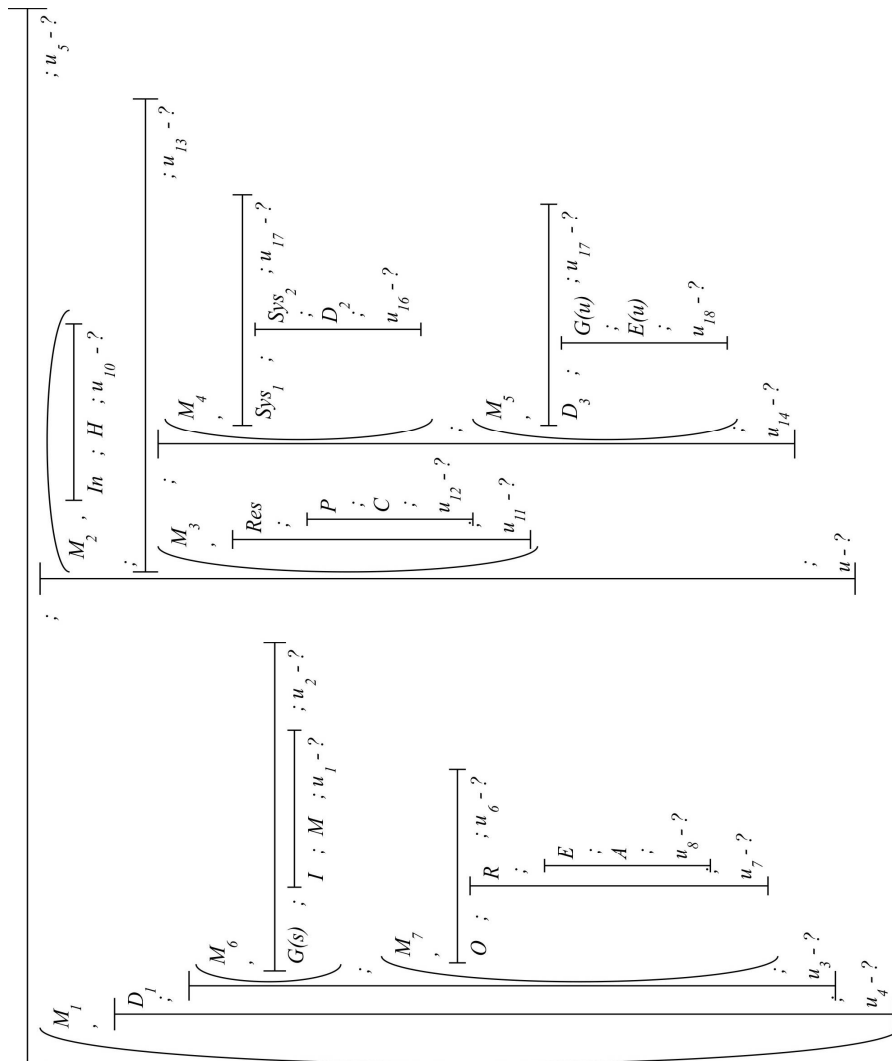
Ці модулі містять такі операції, як відображення інформації про вибраний знак операції чи унітерм. Модуль введення нових параметрів знаку операцій чи унітермів.

Модуль генерування унітерму містить такі модулі:

- модуль генерації унітермів – створення унітермів;

- модуль редагування унітермів – зміна параметрів унітермів.

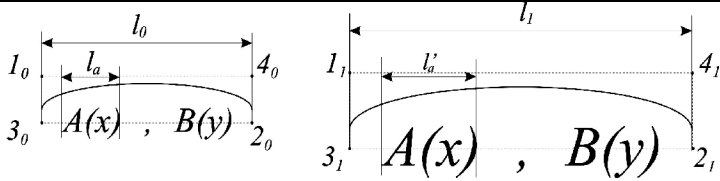
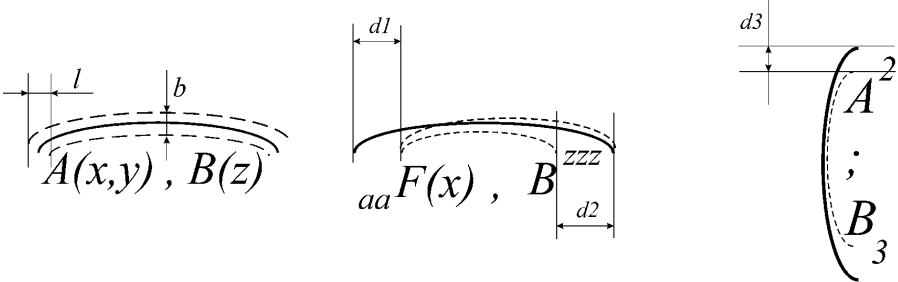
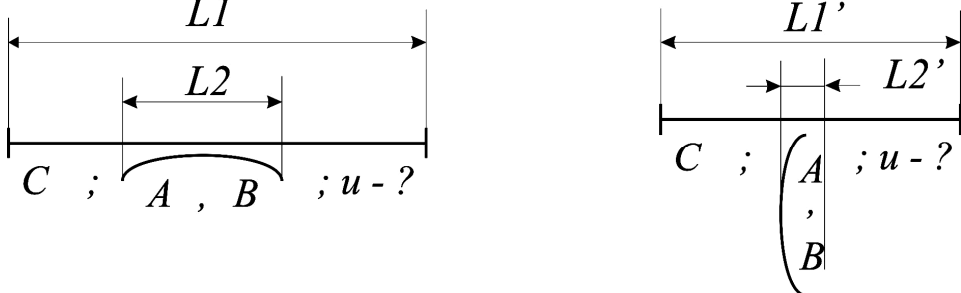
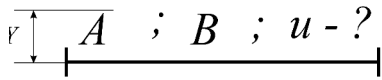
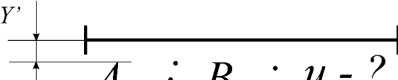
Після синтезу секвенцій та елімінувань і мінімізації алгоритму отримаємо такий алгоритм редактора:



Одним з основних процесів редактора формул алгебри алгоритмів є процес адаптації.

Адаптацією називатимемо процес зміни розмірів знаків операцій залежно від розмірів, розташування (згори/знизу), орієнтації (вертикальна/горизонтальна), кегля, верхніх та нижніх індексів унітермів, а також знаків операцій алгебри алгоритмів.

Задачі адаптації

Задача адаптації	Пояснення
Адаптація залежно від розмірів шрифту	 <p>де l_0 – довжина формули до адаптації, l_a – довжина вкладеної формули; l_1 – довжина формули після адаптації; l'_a – довжина вкладеної формули після зміни кегля.; $1_0, 2_0, 3_0, 4_0$ – точки побудови знаку операції до зміни розмірів знаку; $1_1, 2_1, 3_1, 4_1$ – точки побудови знаку операції після зміни розмірів знаку;</p>
Адаптація з врахуванням індексів	 <p>де l – діапазон зміни розміру знаку операції по горизонталі; b – діапазон зміни розташування знаку операції по вертикалі; $d1$ – діапазон зміни координат розташування формули по горизонталі; $d2$ – діапазон зміни розмірів формули по горизонталі; $d3$ – діапазон зміни координат розташування формули по вертикалі;</p>
Адаптація з врахуванням наявності вертикальної або горизонтальної орієнтації знаків операцій	 <p>де $L1$ – довжина формули, $L2$ – довжина вкладеної формули, $L1'$ – довжина формули після адаптації, $L2'$ – довжина формули після зміни орієнтації;</p>
Адаптація залежно від розташування унітермів	<div style="display: flex; justify-content: space-around;"> <div data-bbox="480 1491 895 1615"> <p>1) згори</p>  </div> <div data-bbox="983 1491 1398 1615"> <p>2) знизу</p>  </div> </div> <p>де Y – відстань по висоті від знаку операції до унітерму при розташуванні згори; Y' – відстань за висотою від знаку операції до унітерму при розташуванні знизу.</p>

Процес адаптації формул абстрактних алгоритмів є складовою процесу набору формул. Цей процес виконують після здійснення користувачем будь-яких змін у редакторі, тобто адаптація є автоматичним процесом, який користувач безпосередньо інструментальними засобами не викликає.

Процес адаптації базового знаку операції секвентування враховує нелінійне співвідношення його довжини і ширини та залежить від розмірів шрифту унітермів, наявності чи відсутності індексів, типу орієнтації знаків операцій та від розташування вкладених формул.

Загальний абстрактний алгоритм процесу адаптації формул алгоритмів

Синтез секвенцій. Секвенція, яка описує процес адаптації за горизонтальною орієнтацією формули алгоритму і за наявності першої вкладеної формули має такий вигляд:

$$S_1 = \overbrace{P_1(l_a, Z_1); F_1; F_2},$$

де Z_1 – позначення алгоритму зміни адаптованих геометричних розмірів за наявності першої вкладеної формули і горизонтальній орієнтації базового знаку операції; F_1 – алгоритм адаптації будь-якої базової формули до вкладеної; F_2 – алгоритм переміщення вкладеної формули до базової.

Процес адаптації при горизонтальній орієнтації формули алгоритму і за наявності другої вкладеної формули описується такою секвенцією:

$$S_2 = \overbrace{P_1(l_a, Z_2); F_1; F_2},$$

де Z_2 – позначення алгоритму зміни адаптованих геометричних розмірів за наявності другої вкладеної формули і горизонтальної орієнтації базового знаку операції.

Секвенція, яка описує процес адаптації за вертикальної орієнтації формули алгоритму і за наявності першої вкладеної формули має такий вигляд:

$$S_3 = \overbrace{P_1(h_a, Z_3); F_1; F_2},$$

де Z_3 – позначення алгоритму зміни адаптованих геометричних розмірів за наявності першої вкладеної формули і вертикальної орієнтації базового знаку операції.

Процес адаптації при вертикальній орієнтації формули алгоритму і за наявності другої вкладеної формули описується такою секвенцією:

$$S_4 = \overbrace{P_1(h_a, Z_4); F_1; F_2},$$

де Z_4 – позначення алгоритму зміни адаптованих геометричних розмірів за наявності другої вкладеної формули і вертикальної орієнтації базового знаку операції.

2. *Синтез елімінавань.* Секвенції S_1 і S_2 елімінуємо за умовою перевірки на наявність першої вкладеної формули (u_2), отримуючи такий вираз:

$$L_1 = \overline{S_1; S_2; u_2?}$$

За умовою перевірки на наявність першої вкладеної формули (u_2) елімінуємо секвенції S_3 і S_4 , отримуючи таку формулу алгоритму:

$$L_2 = \overline{S_3; S_4; u_2?}$$

Елімінування L_1 і L_2 елімінуємо за умовою перевірки на тип орієнтації, в результаті чого отримаємо таку формулу:

$$L_3 = \overline{E_1; E_2; u_1?}$$

Після підстановки секвенцій у відповідні елімінування отримаємо такий абстрактний алгоритм загального алгоритму адаптації формул:

$$\overline{\left(\begin{array}{c} P_1(l_a, Z_1) \\ \vdots \\ F_1 \\ F_2 \end{array} \right); \left(\begin{array}{c} P_1(l_a, Z_2) \\ \vdots \\ F_1 \\ F_2 \end{array} \right); u_2? ; \left(\begin{array}{c} P_1(h_a, Z_3) \\ \vdots \\ F_1 \\ F_2 \end{array} \right); \left(\begin{array}{c} P_1(h_a, Z_4) \\ \vdots \\ F_1 \\ F_2 \end{array} \right); u_2? ; u_1?}$$

На підставі властивості дистрибутивності знаку операції елімінування алгебри алгоритмів вносимо унітерми F_1 і F_2 за знак операції елімінування. Отримаємо таку формулу:

$$A = \left(\begin{array}{c} \overbrace{\left(\begin{array}{c} P_1(l_a, Z_1) ; \\ \vdots \\ P_1(l_a, Z_2) \\ \vdots \\ u_2 - ? \end{array} ; \begin{array}{c} P_1(h_a, Z_3) ; \\ \vdots \\ P_1(h_a, Z_4) \\ \vdots \\ u_2 - ? \end{array} ; u_1 - ? \right)} \\ ; \\ \left(\begin{array}{c} F_1 \\ ; \\ F_2 \end{array} \right) \end{array} \right)$$

Алгоритм адаптації
 будь-якої базової
 формули до вложеної

Абстрактний алгоритм процесу адаптації формул алгоритмів

Синтез секвенцій. Секвенції, які описують процес адаптації, мають такий вигляд:

$$S_5 = \left(\begin{array}{c} P_1(i, i_0) \\ ; \\ K_1 \end{array} \right) \quad S_6 = \left(\begin{array}{c} P_1(i, i_0) \\ ; \\ K_2 \end{array} \right) \quad S_7 = \left(\begin{array}{c} P_1(i, i_0) \\ ; \\ K_3 \end{array} \right) \quad S_8 = \left(\begin{array}{c} P_1(i, i_0) \\ ; \\ K_4 \end{array} \right) \quad S_9 = \left(\begin{array}{c} P_1(i, i_0) \\ ; \\ K_5 \end{array} \right) \quad S_{10} = \left(\begin{array}{c} P_1(i, i_0) \\ ; \\ K_6 \end{array} \right)$$

де $P_1(i, i_0)$ – унітерм присвоєння змінній i значення порядкового номера i_0 будь-якого базового знаку операції; $K_1, K_2, K_3, K_4, K_5, K_6$ – алгоритми адаптації базових знаків паралелення, елімінування, секвентування, циклічного секвентування, циклічного елімінування і циклічного паралелення, відповідно.

2. *Синтез елімінувань.* За умовою перевірки на тип операції: чи він є знаком операції циклічного паралелення (u_6), елімінуємо секвенцію S_{10} і порожній унітерм, отримуючи такий вираз:

$$L_4 = \overbrace{\left(S_{10} ; * ; u_6 - ? \right)}$$

Секвенцію S_9 і елімінування L_4 елімінуємо за умовою перевірки на тип операції: чи він є циклічним елімінуванням (u_5), отримуючи таку формулу алгоритму:

$$L_5 = \overbrace{\left(S_9 ; L_4 ; u_5 - ? \right)}$$

За умовою перевірки на тип операції, чи він є знаком операції циклічного секвентування (u_4) елімінуємо секвенцію S_8 і елімінування L_5 , отримуючи такий вираз:

$$L_6 = \overbrace{\left(S_8 ; L_5 ; u_4 - ? \right)}$$

Секвенцію S_7 і елімінування L_6 елімінуємо за умовою перевірки на тип операції: чи він є секвентуванням (u_3), отримуючи таку формулу алгоритму:

$$L_7 = \overbrace{\left(S_7 ; L_6 ; u_3 - ? \right)}$$

За умовою перевірки на тип операції, чи він є знаком операції елімінування (u_2), елімінуємо секвенцію S_6 і елімінування L_7 , отримуючи такий вираз:

$$L_8 = \overbrace{\left(S_6 ; L_7 ; u_2 - ? \right)}$$

Секвенцію S_5 і елімінування L_8 елімінуємо за умовою перевірки на тип операції: чи він є паралеленням (u_1), отримуючи таку формулу алгоритму:

$$L_9 = \overbrace{\left(S_5 ; L_8 ; u_1 - ? \right)}$$

Після підстановки секвенцій у відповідні елімінування отримаємо таку формулу алгоритму:

$$\left(\begin{array}{c} P_1(i, i_0) \\ ; \\ K_1 \end{array} \right) ; \overbrace{\left(\begin{array}{c} P_1(i, i_0) \\ ; \\ K_2 \end{array} ; \overbrace{\left(\begin{array}{c} P_1(i, i_0) \\ ; \\ K_3 \end{array} ; \overbrace{\left(\begin{array}{c} P_1(i, i_0) \\ ; \\ K_4 \end{array} ; \overbrace{\left(\begin{array}{c} P_1(i, i_0) \\ ; \\ K_5 \end{array} ; \overbrace{\left(\begin{array}{c} P_1(i, i_0) \\ ; \\ K_6 \end{array} ; * ; u_6 - ? \right)} ; u_5 - ? \right)} ; u_4 - ? \right)} ; u_3 - ? \right)} ; u_2 - ? \right)} ; u_1 - ? \right)$$

На підставі властивості дистрибутивності знаку операції елімінування алгебри алгоритмів, виносимо унітерм $P_1(i, i_0)$ за знак операції елімінування, в результаті чого отримуємо таку формулу:

$$F_i = \left(P_1(i, i_0) \right. \\ \left. \left[K_1 ; \left[K_2 ; \left[K_3 ; \left[K_4 ; \left[K_5 ; \left[K_6 ; * ; u_6^{-?} \right] ; u_5^{-?} \right] ; u_4^{-?} \right] ; u_3^{-?} \right] ; u_2^{-?} \right] ; u_1^{-?} \right] \right)$$

Формула алгоритму базового знаку операції секвентування

$$K_3 = \left(\begin{array}{l} P_1(X_1, x_1) \\ ; \\ P_1(Y_1, y_1) \\ ; \\ \left(\begin{array}{l} P_1(X_2, x_1 + l_d) \\ ; \\ P_1(Y_2, y_1 + f(l_d, c_1)) \end{array} \right) ; \left(\begin{array}{l} P_1(X_2, x_1 + f(h_a, c_1)) \\ ; \\ P_1(Y_2, y_1 + h_d) \end{array} \right) ; u_1^{-?} \\ ; \\ \left(\begin{array}{l} P_1(X_3, x_1) \\ ; \\ P_1(Y_3, y_1 + f(l_d, c_1)) \end{array} \right) ; \left(\begin{array}{l} P_1(X_3, x_1 + f(h_a, c_1)) \\ ; \\ P_1(Y_3, y_1) \end{array} \right) \\ ; \\ \left(\begin{array}{l} P_1(X_4, x_1 + l_d) \\ ; \\ P_1(Y_4, y_1) \end{array} \right) ; \left(\begin{array}{l} P_1(X_4, x_4) \\ ; \\ P_1(Y_4, y_1 + h_d) \end{array} \right) \end{array} \right)$$

Абстрактний алгоритм процесу адаптації базового знаку операції секвентування

Процес обчислення ширини першого унітерму та його індексів повинен виконуватися після набору чи редагування користувачем унітерму, тобто цей процес є автоматичним, який користувач безпосередньо інструментальними засобами не викликає.

Схему алгоритму обчислення геометричних розмірів унітерму наведено на рис. 3.

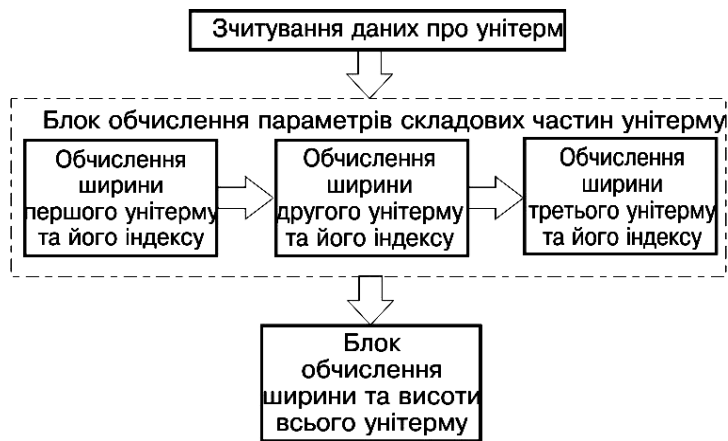


Рис. 3. Схеми алгоритму обчислення геометричних розмірів унітерму

Блоком обчислення параметрів складових нетривіального унітерму описано обчислення ширини і висоти їхніх індексів (верхніх і нижніх), а також ширини і висоти самих складових частин із врахуванням обчислених ширин і висот їхніх індексів.

У блоці обчислення ширини та висоти всього унітерму описано обчислення ширини та висоти унітерму з врахуванням параметрів його складових.

Обчислення геометричних розмірів унітерму описується такою формулою:

$$\left(\begin{array}{c} \left(\begin{array}{c} P_1(h_t, Ht) \\ ; \\ P_1(w_t, Wt) \end{array} \right) ; \left(\begin{array}{c} P_1(h_t, c) ; u_4 - ? \\ ; \\ P_1(w_t, c) \end{array} \right) \\ ; \\ \left(\begin{array}{c} \left(\begin{array}{c} P_1(h_v, Hv) \\ ; \\ P_1(w_v, Wv) \end{array} \right) ; \left(\begin{array}{c} P_1(h_v, c) ; u_3 - ? \\ ; \\ P_1(w_v, c) \end{array} \right) \\ ; \\ \left(\begin{array}{c} P_1(H, c) \\ ; \\ P_1(W, c) \end{array} \right) \end{array} \right) \\ ; \\ \left(\begin{array}{c} \left(\begin{array}{c} P_1(h_d, Hd) \\ ; \\ P_1(w_d, Wd) \end{array} \right) ; \left(\begin{array}{c} P_1(h_d, c) ; u_2 - ? \\ ; \\ P_1(w_d, c) \end{array} \right) \\ ; \\ \left(\begin{array}{c} P_1(W, w_t + w_v) ; P_1(W, w_t + w_d) ; u_1 - ? \end{array} \right) \end{array} \right)$$

Аналогічно синтезується і на основі властивості дистрибутивності операції елімінування мінімізується опис процесів обчислення ширини двох наступних тривіальних унітермів. Обчислення ширини та висоти всього унітерму описується такою формулою:

$$\left(\begin{array}{c} \left(\begin{array}{c} P_1(w_t, \sum_{i=1}^{i=3} w_t^i) \\ ; \\ P_1(w_v, \sum_{i=1}^{i=3} w_v^i) \\ ; \\ P_1(w_d, \sum_{i=1}^{i=3} w_d^i) \end{array} \right) \\ ; \\ \left(\begin{array}{c} \left(\begin{array}{c} P_1(W, w_t + w_v) ; P_1(W, w_t + w_d) ; u_1 - ? \end{array} \right) \\ ; \\ \left(\begin{array}{c} P_1(H_v, c) ; P_1(H_v, H_z) ; u_2 - ? \end{array} \right) \\ ; \\ \left(\begin{array}{c} P_1(H_d, c) ; P_1(H_d, H_z) ; u_3 - ? \end{array} \right) \end{array} \right)$$

Блок-схема обчислення ширини вкладеної формули абстрактного алгоритму має такий вигляд:

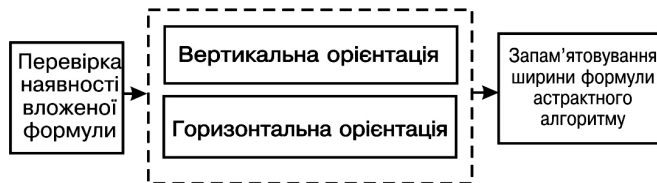
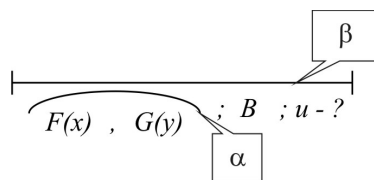


Рис. 4. Блок-схема обчислення ширини формули

Обчислення ширини формули описується формулою:

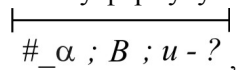
$$L_1 = \left(\begin{array}{c} \left(\begin{array}{c} P_1(W, W_0) \\ ; \\ P_1(W, W_0 + W_a) \end{array} \right) ; \left(\begin{array}{c} P_1(W, W_0) ; u_1 - ? \\ ; \\ P_1(W, W_0 + W_b) \end{array} \right) \end{array} \right)$$

Одним з основних процесів редактора формул алгоритмів є процеси згорнання та розгорнання формул. Процес згорнання формули абстрактного алгоритму має виконуватись так: вибрати формулу абстрактного алгоритму; вибрати на панелі інструментів інструмент “Згорнути чи розгорнути формулу абстрактного алгоритму”. Система автоматично виконає згорнання формули абстрактного алгоритму. Розгорнання формули абстрактного алгоритму має виконуватись за такою послідовністю: вибрати необхідну формулу абстрактного алгоритму (в вигляді унітерму); вибрати на панелі інструментів інструмент “Згорнути чи розгорнути формулу абстрактного алгоритму”. Редактор абстрактних алгоритмів автоматично виконає процес розгорнання формули. На прикладі проілюструємо процес згорнання формули абстрактного алгоритму. Нехай дано формулу абстрактного алгоритму:



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

Після згорнання секвенції отримаємо таку формулу:



де $\#_\alpha$ – умовне позначення формули згорнутої секвенції.

Згорнання – процес перетворення формули алгоритму до її умовного позначення (наприклад, унітерм $\#_\alpha$).

Синтез секвенцій. Абстрактний алгоритм згорнання формули передбачає такі етапи:



Рис. 5. Блок-схема процесу згорнання

Секвенція створення згорнутої формули (S_1) матиме вигляд:

$$S_1 = \left(\begin{array}{c} \overbrace{P_1(T, f_1(c_r, N)) ; P_1(i, f_2(i)) ; P_1(i, T)} \\ ; \\ \overbrace{\left(\begin{array}{c} P_1(X_1, x_1) \\ ; \\ P_1(Y_1, y_1) \end{array} ; \begin{array}{c} P_1(X_2, f_3(X_1, i_1)) \\ ; \\ P_1(X_2, f_4(Y_1, i_1)) \end{array} \right)} \\ ; \\ \overbrace{P_1(H, f_5(Y_1, Y_2)) ; P_1(W, f_5(X_1, X_2))} \\ ; \\ \overbrace{P_1(i_n, i) ; P_1(t, c_2)} \\ ; \\ P_1(F_1, F_{id}) \end{array} \right)$$

Переміщення розгорнутої формули в невидиму область робочого поля редактора супроводжується такими процесами:

- присвоєння координатам формули до згортання значення $c3$ для переміщення в невидиму зону робочого поля $P_1(X_n^{id}, c3)$ і $P_1(Y_n^{id}, c3)$, де X_n^{id} і Y_n^{id} – змінні опису координат для формули до згортання, присвоєння першій вкладеній формулі ідентифікаційного номера згорнутої формули описується унітермом $P_1(A_j, i_n)$. Ці процеси описуються секвенцією S_2 :

$$S_2 = \overbrace{P_1(A_j, i_n) ; P_1(X_n^{id}, c3) ; P_1(Y_n^{id}, c3)}$$

- Секвенція, якою описується процес присвоєння координатам значення $c3$ для переміщення розгорнутої формули в невидиму область робочого поля редактора і присвоєння другій вкладеній формулі ідентифікаційного номера згорнутої формули, описується унітермом $P_1(B_j, i_n)$, має такий вигляд:

$$S_3 = \overbrace{P_1(B_j, i_n) ; P_1(X_n^{id}, c3) ; P_1(Y_n^{id}, c3)}$$

Синтез елімінувань. Абстрактний алгоритм згортання формули передбачає такі елімінування:

- Секвенції S_2 і S_3 елімінуємо за умовою перевірки, чи згорнута формула є першою вкладеною (u_1), отримаємо формулу:

$$L_1 = \overbrace{S_2 ; S_3 ; u_1 - ?}$$

- Елімінування L_1 і порожній унітерм $*K$ елімінуються за умовою перевірки, чи згорнута формула є вкладеною в базову формулу (u_2), у результаті чого отримаємо таку формулу:

$$L_2 = \overbrace{L_1 ; *K ; u_2 - ?}$$

Після синтезу всіх секвенцій та елімінувань отримуємо такий абстрактний алгоритм згортання формул:

$$\left(\begin{array}{c} \overbrace{\left(\begin{array}{c} P_1(T, f_1(c_r, N)) ; P_1(i, T) \\ ; \\ P_1(i, f_2(i)) \end{array} \right)} \\ ; \\ \overbrace{\left(\begin{array}{c} P_1(X_1, x_1) \\ ; \\ P_1(Y_1, y_1) \end{array} ; \begin{array}{c} P_1(X_2, f_3(X_1, i_1)) \\ ; \\ P_1(X_2, f_4(Y_1, i_1)) \end{array} \right)} \\ ; \\ \overbrace{P_1(H, f_5(Y_1, Y_2)) ; P_1(W, f_5(X_1, X_2))} \\ ; \\ \overbrace{P_1(i_n, i) ; P_1(t, c_2) ; P_1(F_1, F_{id})} \\ ; \\ \overbrace{S_j = i} \\ ; \\ \overbrace{\left(\begin{array}{c} P_1(A_j, i_n) ; P_1(B_j, i_n) ; u_1 - ? \\ ; \\ P_1(X_n^{id}, c3) \\ ; \\ P_1(Y_n^{id}, c3) \end{array} \right)} ; *K ; u_2 - ? \\ ; \\ c_{j=i} \end{array} \right)$$

Процес розгортання формул алгоритмів

Розгортання – процес перетворення умовного позначення формули алгоритму (наприклад, унітерм #_ α до формули алгоритму.

Нижче схематично показано блок-схему процесу розгортання формул алгоритмів:



Рис. 6. Блок-схема процесу розгортання формул алгоритмів

Синтез секвенцій. Абстрактний алгоритм розгортання формули передбачає такі процеси:

- присвоєння назви згорнутої формули $P_1(T, id_i)$, де T – змінна, якій присвоюється назва згорнутої формули (id_i);
- присвоєння порядкового номера знайденої згорнутої формули $P_1(ID, f_1(T))$, де ID – порядковий номер знайденої згорнутої формули, $f_1(T)$ – функція пошуку порядкового номера формули за заданою назвою формули;
- переміщення розгорнутої формули в задану точку $P_1(ID, id_x, id_y)$, де id_x, id_y – нові координати розгорнутої формули;
- присвоєння нульової константи унітерму згорнутої формули $P_1(id, c)$, де id – порядковий номер згорнутої формули, c – нульова константа.

Ці унітерми описуються такою секвенцією:

$$S_1 = \left(\begin{array}{l} P_1(ID, f_1(T)) \\ ; \\ P_2(ID, id_x, id_y) \\ ; \\ P_1(id, c) \end{array} \right)$$

Синтез елімінувань. Абстрактний алгоритм розгортання формули передбачає такі елімінування:

- унітерм $P_1(B_j, f_1(T))$ і порожній унітерм з ознакою закінчення пошуку (K) елімінуються за умовою перевірки, чи згорнута формула є першою вкладеною в базову формулу (u_1):

$$L_1 = \left| \begin{array}{l} P_1(A_j, f_1(T)) \quad ; \quad P_1(B_j, f_1(T)) \quad ; \quad u_1 - ? \end{array} \right|$$

- унітерм $P_1(A_j, f_1(T))$ і елімінування L_1 елімінуються за умовою перевірки, чи згорнута формула є першою вкладеною в базову формулу (u_2):

$$L_2 = \left| \begin{array}{l} L_1 \quad ; \quad *K \quad ; \quad u_2 - ? \end{array} \right|$$

Після синтезу всіх секвенцій та елімінувань отримаємо такий абстрактний алгоритм розгортання формул абстрактних алгоритмів:

$$\left(\begin{array}{l} P_1(T, id_i) \\ ; \\ \bigcirc_{j=i} \\ \left(\begin{array}{l} P_1(A_j, f_1(T)) \quad ; \quad P_1(B_j, f_1(T)) \quad ; \quad u_1^-? \\ ; \\ c_{j=1} \end{array} \right) ; *K \quad ; \quad u_2^-? \\ ; \\ \left(\begin{array}{l} P_1(ID, f_1(T)) \\ ; \\ P_2(ID, id_x, id_y) \\ ; \\ P_1(id, c) \end{array} \right) \end{array} \right)$$

Висновки та перспективи подальших розвідок

1. Синтезована, мінімізована і досліджена математична процесу адаптації, згортання і розгортання формул алгебри алгоритмів.

2. Абстрактним алгоритмом структури даних редактора забезпечено повноцінний опис формул алгебри алгоритмів.

1. Овсяк В., Бритковський В., Овсяк О., Овсяк Ю. Синтез і дослідження алгоритмів комп'ютерних систем. – Львів, 2004. – 276 с. 2. Овсяк В. АЛГОРИТМИ: методи побудови, оптимізації, дослідження вірогідності. – Львів: Світ, 2001. – 160 с. 3. Василюк А. С. Адаптивний синтез формул алгоритмів / А. С. Василюк, Т. М. Басюк // Інформаційні системи та мережі: Вісник Нац. ун-ту “Львівська політехніка”. – 2012.– № 743. – С. 35–43. 4. Василюк А. С. Інтелектуальний аналіз геометричних розмірів горизонтального циклічного секвенування / А. С. Василюк // Тези доповідей міжнародної науково-технічної конференції „Обчислювальний інтелект (результати, проблеми, перспективи) (ОІ-2013)”. – Черкаси (10–13 травня): 2013. – С. 429–430. 5. Овсяк В., Василюк А. Принцип побудови підсистеми редагування формул абстрактних алгоритмів // Комп'ютерні технології друкарства – Львів: УАД, – 2004. – № 12. – С. 137–146. 6. Василюк А. С. Математична модель адаптації формул алгоритмів / А. С. Василюк // Тези доповідей міжнародної науково-практичної конференції “Інформаційні управляючі системи та технології” (ІУСТ-ОДЕСА-2013).