УДК 004.421

А.В. Овсяк, В.К. Овсяк

Модифицированная алгебра алгоритмов и инструментальные средства обработки формул алгебры алгоритмов

Для решения задач повышения эффективности обработки структур данных и формул алгоритмов аксиоматическим методом определены алгебры секвенционных алгоритмов первого и второго порядков, использование которых показано на примерах. Описаны эффективные инструментальные средства компьютерного синтеза и оптимизации формул алгоритмов.

The algebra of sequential algorithms of the first and second order are defined to solve the problems of improvement of the efficiency in processing the data structures and the formulas of algorithms by means of the axiomatic method. The application of the algebras is illustrated by means of the examples. The derived effective tools of computer aided synthesis and the optimization of formulas of algorithms are briefly described.

Для розв'язання задач підвищення ефективності опрацювання структур даних і формул алгоритмів аксіоматичним методом означено алгебри секвенційних алгоритмів першого та другого порядків, застосування яких проілюстровано прикладами. Описано створені ефективні інструментальні засоби комп'ютерного синтезу та оптимізації формул алгоритмів.

Введение и формулирование задач исследования. Информационные технологии интенсивно развиваются и нашли практическое применение в различных сферах человеческой деятельности. Математической основой информационных технологий есть теория алгоритмов. Наиболее известными и наиболее часто используемыми методами описания алгоритмов есть вербальный и блок-схемный методы. Кроме них известны и другие методы, которые по используемым ими средствам можно разбить на две группы. Это методы, не использующие алгебраическую методологию и методы, которые ее используют. Исторически неалгебраические методы появились первыми. К ним, кроме упомянутых вербального и блок-схемного представления алгоритмов, принадлежат и такие, как методы рекурсивных функций [1], исчисления лямбда [2], виртуальных машин Тьюринга [3] и Поста [4], алгоритмов Маркова [5], машин Колмогорова [6], Шенхаге [7], Ахо–Улльмана–Хопкрофта [8] и универсальных алгоритмов Криницкого [9].

Известно [10, 11], что алгебраическими методами описания алгоритмов служит система алгебраических алгебр Глушкова, модифицированная система алгебраических алгебр (алгебра алгоритмов Цейтлина), алгебры Дейкстры, Калужнина и Янова, а каждая из них имеет соответствующий клон. Алгебра алгоритмики – это двухуровневая система с неинтерпретированными схемами верхнего уровня и необходимостью построения прикладных алгебр на нижнем уровне [12].

Алгебраические в сравнении с неалгебраическими методами описания алгоритмов имеют то преимущество, что над формулами алгоритмов могут быть автоматически выполнены тождественные преобразования на основе свойств операций, как над произвольными математическими выражениями. Результаты этих преобразований – уменьшение затрат памяти на сохранение и времени на выполнение.

Исследованием [13] начато формирование метода описания алгоритмов, который базируется на идее создания операций упорядочения на базе индексов. Метод со временем развит до прикладной алгебры секвенционных алгоритмов [14]. Многолетняя практика использования прикладной алгебры секвенционных алгоритмов в разных областях науки и техники показала целесообразность упрощения алгоритмов, что обеспечивает сокращение затрат на реализацию алгоритмов. Результаты практики использования послужили основанием для развития специализированной алгебры секвенционных алгоритмов, рассмотренной в настоящей статье.

Алгебра секвенционных алгоритмов первого порядка

Систему обозначений образуют: $\frown$ – секвенцирование; $\vdash\!\!\!\!\dashv$ – элиминирование; $\overline{}$ – реверсирование; $\ulcorner\!\urcorner$ – параллелирование; переменные: $A, B, C, \ldots$; логические переменные: $u, u_0, \ldots, u_j, u_j^i, \ldots$; $*$ – пустой знак; разделители переменных (запятая, точка с запятой и двоеточие); индексы порядка α и β.

В определениях двоеточие как разделитель может быть заменено запятой или точкой с запятой.

Определение 1. Операция, обладающая свойствами: идемпотентности: $\overparen{X, X}=X$; поглощения пустого знака: $\overparen{*: X}=X$; коммутативности: $\overparen{X, Y}=\overparen{Y, X}$; ассоциативности: $\overparen{\overparen{X, Y}, Z}=\overparen{X, \overparen{Y, Z}}$; поглощения упорядоченных переменных: $\overparen{\overparen{X; Y}; \overparen{Z; Y}}=\overparen{\overparen{X, Z}; Y}$, $\overparen{\overparen{X; Y}; \overparen{X; Z}}=\overparen{X; \overparen{Y, Z}}$; – поглощения переменной логическим значением 0: $\overparen{X: 0}=0$; поглощения логического значения 1: $\overparen{X; 1}=X$; образования пар с одинаковыми индексами порядка: $\overparen{X, Y}=\{\{X, \alpha\}, \{Y, \alpha\}\}$; образования пар с различными индексами порядка: $\overparen{X; Y}=\{\{X, \alpha\}, \{Y, \beta\}\}$ называется *секвенцированием*.

Определение 2. Операция, обладающая свойствами: идемпотентности: $\overline{X, X}=X$; поглощения пустого знака: $\overline{*, X}=X$; коммутативности: $\overline{X, Y}=\overline{Y, X}$; ассоциативности: $\overline{\overline{X, Y}, Z}=\overline{X, \overline{Y, Z}}$; выноса переменной: $\overparen{X: \overline{Y, Z}}= \overline{\overparen{X: Y}, \overparen{X: Z}}$, $\overline{\overparen{X: Z}, \overparen{Y: Z}}=\overparen{\overline{X, Y}: Z}$; – поглощения логического значения: $\overline{X: 0}=X$; – образования пар с одинаковыми индексами порядка: $\overline{X, Y}=\{\{X, \gamma\}, \{Y, \gamma\}\}$; образования пар с различными индексами порядка: $\overline{X; Y}=\{\{X, \gamma\}, \{Y, \delta\}\}$ называется *параллелированием*.

Определение 3. Операция, обладающая свойствами: выбора переменной:

$$\overline{X; Y; u\text{–?}} = \begin{cases} X, \text{ если } u = 1, \\ Y, \text{ если } u = 0; \end{cases}$$

$$\overline{X; Y; \ldots Z; w\text{-?}} = \begin{cases} X, \text{ если } (w = v_0) \text{–?}, \\ Y, \text{ если } (w \neq v_0) \& (w = v_1), \\ \ldots \\ Z, \text{ если } (w \neq v_0) \& (w \neq v_1) \& \ldots (w \neq v_{n-1}); \end{cases}$$

где $v_0, v_1, \ldots v_{n-1}$ – значения w; идемпотентности: $\overline{X; X; u\text{-?}}=X$, $\overline{X; X; \ldots X; w\text{-?}}=X$; выбора логической переменной: $\overline{\overline{X; Y; u_1\text{-?}}; \overline{X; Y; u_2\text{-?}}; u_3\text{-?}} = \overline{X; Y; \overline{u_1\text{-?}; u_2\text{-?}; u_3\text{-?}}}$; дистрибутивности: $\overline{\overparen{R: S}; \overparen{R: T}; u\text{-?}} = \overparen{R: \overline{S; T; u\text{-?}}}$, $\overline{\overparen{S: R}; \overparen{T: R}; u\text{-?}} = \overparen{\overline{S; T; u\text{-?}}: R}$, $\overline{\overparen{A:B}; \overparen{A:C}; \ldots \overparen{A:N}; w\text{-?}} = \overparen{A: \overline{B; C; \ldots; N; w\text{-?}}}$, $\overline{\overparen{A:B}; \overparen{C:B}; \ldots; \overparen{N:B}; w\text{-?}} = \overparen{\overline{A; C; \ldots; N; w\text{-?}}: B}$; поглощения переменной: $\overline{X; \overline{Y; Z; u\text{ - ?}}; u\text{-?}} = \overline{X; Z; u\text{-?}}$, $\overline{\overline{X; Y; u\text{ - ?}}; Z; u\text{-?}} = \overline{X; Z; u\text{-?}}$, $\overline{\overline{A; B; \ldots Z; w\text{-?}}; C; K; \ldots I; w\text{-?}} = \overline{A; C; K; \ldots I; w\text{-?}}$, $\overline{A; B; \ldots Z; \overline{Q; L; \ldots I; w\text{-?}}; w\text{-?}} = \overline{A; B; \ldots Z; I; w\text{-?}}$, называется *элиминированием*.

Определение 4. Операция, обладающая свойствами: преобразования секвенцирования: $\overline{\overparen{u_1: u_2}}=\overline{\overline{u_1}:\overline{u_2}}$; преобразования параллелирования: $\overline{\overline{u_1: u_2}}=\overparen{\overline{u_1}: \overline{u_2}}$; двойного преобразования: $\overline{\overline{u}}=u$; отрицания элиминирования: $\overline{\overline{u_1; u_2; u_3\text{-?}}} = \overline{\overline{u_1}; \overline{u_2}; u_3\text{-?}}$, $\overline{\overline{R; S; \ldots; Z; w\text{-?}}} = \overline{\overline{R}; \overline{S}; \ldots; \overline{Z}; w\text{-?}}$; – замены 0: 0 = 1; образования логического значения 0: $\overparen{u, \overline{u}}=0$; связи операций $\overline{X; Y; u\text{-?}} = \overline{\overparen{u; X}, \overparen{\overline{u}; Y}}$, называется *реверсированием*.

Определение 5. Переменные или их значения с приписанными им операциями секвенцирования, элиминирования и параллелирования индексами порядка называются *унитермами*.

Определение 6. Формулой (структурой) есть только такое выражение, для которого это можно показать применением конечного количества раз пп. 1–3:

1. Если X и Y – сменные или унитермы, то $\overparen{X: Y}$ и $\overline{X: Y}$ – формулы;

2. Если $A, B, \ldots, Z$ – переменные, унитермы или формулы, а u, w – унитермы условий, то $\overline{A; B; u\text{-?}}$, $\overline{B; A; u\text{ - ?}}$, $\overline{B; A; \ldots; Z; w\text{ - ?}}$ – формулы;

3. Если A, B, Q, L, ..., T – унитермы или формулы, а u, w – унитермы условий, то $\overline{A}$, $\overline{B}$, $\overline{\overparen{A: B}}$, $\overline{\overline{A: B}}$, $\overline{\overline{A; B; u\text{-?}}}$, $\overline{\overline{B; A; u\text{-?}}}$, – формулы.

Переменные формул могут заменятся переменными или формулами одновременно всюду, где они входят в формулу, в которой выполняется замена.

Алгеброй секвенционных алгоритмов первого порядка есть система операций секвенцирования, элиминирования, параллелирования и реверсирования над переменными, унитермами и формулами.

Пример 1. Пусть переменные X и Y принимают значения из секвенционной области $Q = \overparen{0, 1}$ и упорядочиваются индексами порядка α и β. В этом случае переменным X и Y операциями секвенцирования и параллелирования могут быть приписаны индексы X_α, Y_α, X_β и Y_β. В предложенных определениях операций для большей наглядности формул, индексы упорядочений переменных опущены, а в табл. 1 они заданы явно. Для ориентации упорядочений $\alpha\,\beta$ (возможна противоположная ориентация, а именно $\beta\,\alpha$, реверсивная к предыдущей ориентации) заданы все возможные значения индексированных переменных и некоторые возможные упорядочения.

Элементы секвенционной логики [15]

Определение 7. Операция, обладающая свойствами: идемпотентности: $\&\overparen{u, u} = u$; коммутативности: $\&\overparen{u_1, u_0} = \&\overparen{u_0, u_1}$; образования логического значения: $\&\overparen{u: \overline{u}} = 0$; выэлиминирования операции: $\&\overparen{u: 0} = 0$; ассоциативности: $\&\overparen{u_0, \&\overparen{u_1, u_2}} = \&\overparen{\&\overparen{u_0, u_1}, u_2}$; получения переменной: $\&\overparen{X: 1} = X$, назовем *секвенционной конъюнкцией*.

Определение 8. Операция, обладающая свойствами: идемпотентности: $\vee\overline{u, u} = u$; коммутативности: $\vee\overline{u_1, u_0} = \vee\overline{u_0, u_1}$; образования логического значения: $\vee\overline{u, \overline{u}} = 1$; выэлиминирования операции: $\vee\overline{u, 1} = 1$; выэлиминирования значения: $\vee\overline{u, 0} = u$; ассоциативности: $\vee\overline{u_0, \vee\overline{u_1, u_2}} = \vee\overline{\vee\overline{u_0, u_1}, u_2}$; дистрибутивности: $\vee\overline{\&\overparen{u: u_0}, \&\overparen{u: u_1}} = \&\overparen{u: \vee\overline{u_0, u_1}}$, называется *секвенционной дизъюнкцией*.

Определение 9. Операция, обладающая свойствами: получения из секвенционной конъюнкции секвенционной дизъюнкции: $\&\overparen{\overline{u_0}: u_1} = \vee\overline{\overline{u_0}: \overline{u_1}}$; образования логического значения $0: \overline{0} = 1$; двойного преобразования: $\overline{\overline{u}} = u$, называется *секвенционным инвертированием*.

Секвенционная логика – это система операций секвенционных конъюнкций, дизъюнкции и инвертирования над секвенционными логическими значениями и переменными.

В табл. 1 представлены истинностные значения секвенционных операций конъюнкции и дизъюнкции над упорядоченными логическими переменными и значениями.

Теорема 1. Имеется формула:

$$\overline{X; Y; u \text{ - } ?} = \vee\overline{\&\left(\begin{matrix} u, \\ ; \\ Y \end{matrix}\right. \&\left(\begin{matrix} \overline{u} \\ ; \\ Y. \end{matrix}\right.}$$

Доказательство. Пусть секвенционная переменная u имеет значение 1. Тогда из конъюнкции $\&\overparen{u; X}$ заменой переменной u ее значением 1 получим формулу $\&\overparen{1; X}$, из которой на основании свойства получения переменной операции конъюнкции выведем X. Из секвенционной конъюнкции $\&\overparen{u; X}$ подстановкой вместо переменной u ее значения 1 выводим формулу $\&\overparen{\overline{1}; X}$. Отсюда на основании преобразования логического значения операции секвенционного инвертирования выводим $\&\overparen{0; X}$, откуда, использовав свойство выэлиминирования операции секвенционной конъюнкции, получаем ноль. Подставляя полученные значения секвенционных конъюнкций в секвенционную дизъюнкцию формулы утверждения, выводим выражение $\vee\overline{X; 0}$. Отсюда, используя свойство выэлеминирования значения операции секвенционной дизъюнкции, получаем X. Полученная переменная совпадает с переменной определения операции элиминирования на основании свойства выбора переменной по логической переменной. При $u = 0$ аналогично выводится совпадение переменной Y. Теорема доказана.

Пример 2. Известен [10] «пузырьковый» алгоритм для сортирования массива. Рассмотрим, как сортирование массива 5 2 3 1. 4 может быть описано с использованием алгебры секвенционных алгоритмов. Цифра 1 с точкой означает, что это – число десятичной системы.

Т а б л и ц а 1. Упорядочения операциями секвенцирования, параллелирования и операции секвенционной логики

№	Индексы				$\overset{\frown}{X_\alpha, Y_\alpha}$	$\overline{\overset{\frown}{X_\alpha, Y_\alpha}}$	$\overline{\overline{X_\alpha}, \overline{Y_\alpha}}$	$\& \overset{\frown}{X; Y}$	$\& \overset{\frown}{Y; X}$	$\vee \overline{X, Y}$	$\& \overline{\overset{\frown}{X; Y}}$	$\vee \overline{\overline{X}, \overline{Y}}$
	α		β									
	Переменные											
	X_α	Y_α	X_β	Y_β								
0	0	0	0	0	0	1	1	0	0	0	1	1
1	0	0	0	1	0	1	1	0	0	1	1	1
2	0	0	1	0	0	1	1	0	0	0	1	1
3	0	0	1	1	0	1	1	0	0	1	1	1
4	0	1	0	0	0	1	1	0	0	0	1	1
5	0	1	0	1	0	1	1	0	0	1	1	1
6	0	1	1	0	0	1	1	0	1	0	1	1
7	0	1	1	1	0	1	1	0	1	1	1	1
8	1	0	0	0	0	1	1	0	0	1	1	1
9	1	0	0	1	0	1	1	1	0	1	0	0
10	1	0	1	0	0	1	1	0	0	1	1	1
11	1	0	1	1	0	1	1	1	0	1	0	0
12	1	1	0	0	1	0	0	0	0	1	1	1
13	1	1	0	1	1	0	0	1	0	1	0	0
14	1	1	1	0	1	0	0	0	1	1	1	1
15	1	1	1	1	1	0	0	1	1	1	0	0

Сначала опишем этот массив в виде формулы алгебры секвенционных алгоритмов с использованием операции секвенцирования, что обусловлено одновременным наличием всех элементов массива, которые разделим запятой: $\overset{\frown}{5, \overset{\frown}{2, \overset{\frown}{3, \overset{\frown}{1., 4}}}}$. Применив к ней свойство ассоциативности операции секвенцирования, выведем формулу $\overset{\frown}{\overset{\frown}{5, 2}, \overset{\frown}{3, \overset{\frown}{1., 4}}}$, к которой применим свойство коммутативности операции секвенцирования, что дает $\overset{\frown}{\overset{\frown}{2, 5}, \overset{\frown}{3, \overset{\frown}{1., 4}}}$. К полученному выражению еще семь раз применим свойства ассоциативности и коммутативности, в результате чего получим отсортированный массив $\overset{\frown}{1., \overset{\frown}{2, \overset{\frown}{3, \overset{\frown}{4, 5}}}}$. Описанные преобразования тождественны, что можно записать как: $\overset{\frown}{5, \overset{\frown}{2, \overset{\frown}{3, \overset{\frown}{1., 4}}}} = \overset{\frown}{\overset{\frown}{5, 2}, \overset{\frown}{3, \overset{\frown}{1., 4}}} = \overset{\frown}{\overset{\frown}{2, 5}, \overset{\frown}{3, \overset{\frown}{1., 4}}} = \ldots =$
$= \overset{\frown}{1., \overset{\frown}{2, \overset{\frown}{3, \overset{\frown}{4, 5}}}}$.

Рассмотренный пример иллюстрирует возможности использования алгебры секвенционных алгоритмов для описания и преобразования структур данных.

Алгебра секвенционных алгоритмов второго порядка. Систему обозначений образуют: знаки алгебры алгоритмов первого порядка; циклического секвенцирования ($\not\subset$); циклического элиминирования ($\not\supset$); циклического параллелирования ($\varnothing$); знака равенства (=); *функциональные переменные* $F(a)$, $G\overset{\frown}{a; c}$, $H\overset{\frown}{k; \overset{\frown}{p, r}}$, …, зависящие от *индивидных* переменных a, b, …, упорядоченных операциями секвенцирования и параллелирования.

Операция равенства имеет известное определение ($X = X$, если $X = Y$, то $Y = X$, если $X = Y$, а $Y = Z$, то $X = Z$).

Определение 10. Операция со свойствами: реверсирования: $\overline{\not\subset x F(x)} = \varnothing x \overline{F(x)}$; секвенцирования: $\not\subset x F(x) = \overset{\frown}{F(i): \overset{\frown}{F(j): \overset{\frown}{F(k): \ldots}}}$, для $x \in Q = \overset{\frown}{i: \overset{\frown}{j: \overset{\frown}{k: \ldots}}}$; пустого цикла: $\not\subset x *_x = *$, назовем *циклическим секвенцированием*.

Индивидные переменные, расположенные под знаками циклических операций, названы *связанными* этими операциями, а не связанные переменные – *свободными*.

Определение 11. Операцию со свойствами: реверсирования: $\overline{\varnothing x F(x)} = \not\subset x \overline{F(x)}$; параллелирования: $\varnothing x F(x) = \overline{F(i): \overline{F(j): \overline{F(k): \ldots}}}$, для $x \in Q = \overset{\frown}{i: \overset{\frown}{j: \overset{\frown}{k: \ldots}}}$; пустого цикла: $\varnothing x *_x = *$, назовем *циклическим параллелированием*.

Определение 12. Операцию со свойствами: элиминирования: $\not\supset u_x F(x) =$
$= F(i); F(j); F(k); \ldots u_k - ?; u_j - ?; u_i - ?$, для $x \in Q =$

$=\widehat{i:\widehat{j:\widehat{k:\ldots}}}$; пустого цикла: $\not\supset x\,*_x=*$, назовем *циклическим элиминированием*.

Определение 13. Функциональные переменные, упорядоченные операциями секвенцирования, элиминирования, параллелирования, циклическим секвенцированием, элиминированием и параллелированием – функциональные унитермы.

Определение 14. Алгоритм это выражение, для которого необходимо применить пп. 1–5 конечное количество раз:

1. Если D, K – переменные или функциональные унитермы или формулы, то $=\widehat{D;K}$ и $D=K$ – алгоритмы.

2. Если $F\widehat{a:\widehat{b:\ldots}}$ и $P\widehat{c:\widehat{d:\ldots}}$ – функциональные унитермы, зависящие от одной или нескольких индивидных переменных, то $\widehat{F\widehat{a:\widehat{b:\ldots}}:P\widehat{c:\widehat{d:\ldots}}}$ и $\overline{F\widehat{a:\widehat{b:\ldots}}:P\widehat{c:\widehat{d:\ldots}}}$ – алгоритмы.

3. Если $A\widehat{a:\widehat{b:\ldots}}$, $B\widehat{c:\widehat{d:\ldots}}$, …, $Z\widehat{k:\widehat{f:\ldots}}$ – функциональные переменные или алгоритмы, а u, w – унитермы условий, то $\overline{A\widehat{a:\widehat{b:\ldots}};B\widehat{c:\widehat{d:\ldots}};u\text{-?}}$, $\overline{B\widehat{c:\widehat{d:\ldots}};A\widehat{a:\widehat{b:\ldots}};u\text{-?}}$ и $\overline{A\widehat{a:\widehat{b:\ldots}};B\widehat{c:\widehat{d:\ldots}};\ldots;Z\widehat{k:\widehat{f:\ldots}};w\text{-?}}$ – алгоритмы.

4. Если $A\widehat{a:\widehat{b:\ldots}}$, $B\widehat{c:\widehat{d:\ldots}}$, $Q\widehat{q:\widehat{p:\ldots}}$, $L\widehat{g:\widehat{h:\ldots}}$, …, $T\widehat{r:\widehat{k:\ldots}}$ – логические функциональные унитермы или алгоритмы, а u, w – унитермы условий, то $\overline{A\widehat{a:\widehat{b:\ldots}}}$, $\overline{B\widehat{c:\widehat{d:\ldots}}}$, $\widehat{A\widehat{a:\widehat{b:\ldots}}:B\widehat{c:\widehat{d:\ldots}}}$, $\overline{\overline{A\widehat{a:\widehat{b:\ldots}}:B\widehat{c:\widehat{d:\ldots}}}}$, $\overline{A\widehat{a:\widehat{b:\ldots}};B\widehat{c:\widehat{d:\ldots}};u\text{-?}}$, $\overline{\overline{B\widehat{c:\widehat{d:\ldots}};A\widehat{a:\widehat{b:\ldots}};\bar{u}\text{-?}}}$ и $\overline{\overline{Q\widehat{q:\widehat{p:\ldots}};L\widehat{g:\widehat{h:\ldots}};\ldots,T\widehat{r:\widehat{k:\ldots}};w\text{-?}}}$, $\overline{\overline{Q\widehat{q:\widehat{p:\ldots}};L\widehat{g:\widehat{h:\ldots}};\ldots,T\widehat{r:\widehat{k:\ldots}};\bar{w}\text{-?}}}$ – алгоритмы.

5. Если x, a, … – переменные, $F\widehat{x:\widehat{a:\ldots}}$ – функциональный унитерм или алгоритм, а $F\widehat{x:\widehat{a:\ldots}}$ – логическим функциональным унитермом или алгоритмом, то $\not\subset x\,A\widehat{x:\widehat{a:\ldots}}$, $\not\supset x\,A\widehat{x:\widehat{a:\ldots}}$, $\overline{\varnothing x\,A\widehat{x:\widehat{a:\ldots}}}$, $\overline{\not\subset x\,F\widehat{x:\widehat{a:\ldots}}}$ и $\overline{\not\supset x\,F\widehat{x:\widehat{a:\ldots}}}$, $\overline{\varnothing x\,F\widehat{x:\widehat{a:\ldots}}}$ – алгоритмы.

В алгоритмах могут выполняться замены переменных их значениями, переменными, формулами и алгоритмами одновременно всюду, где заменяемые переменные входят в алгоритм, которым выполняется замена. Индивидные переменные функциональных унитермов могут одновременно во всех вхождениях заменяться их значениями или другими индивидными переменными. Функциональные переменные с несвязанными операциями циклов индивидными переменными могут одновременно во всех вхождениях в алгоритм заменятся другими функциональными унитермами с количеством несвязанных переменных, не меньшим заменяемого функционального унитерма. Связанные операциями циклов индивидные переменные могут одновременно во всех вхождениях заменятся другими несвязанными индивидными переменными.

Алгеброй алгоритмов второго порядка есть система операций секвенцирования, элиминирования, параллелирования, реверсирования, равенства и операций циклов над унитермами.

Пример 3. Использование алгебры секвенционных алгоритмов для синтеза и исследования алгоритмов рассмотрим на примере алгоритма Евклида для вычисления наибольшего общего делителя двух натуральных чисел x и y.

Пусть N – секвенционная область натуральных чисел ($N=\widehat{1,\widehat{2,\ldots}}$). Проверку принадлежности введенных ($\leftarrow$) переменных x и y к натуральным числам запишем как $\in\widehat{x,N}$ и $\in\widehat{y,N}$. Для x и y должно выполняться соотношение $>\widehat{x;y}$. Выполнение этих трех условий обязательно, что запишем так:

$$\left(\begin{array}{l}\leftarrow\widehat{x;a};\\ \overline{\leftarrow\widehat{y;b};\ *K_0;\ \in\widehat{x;N}\text{ - ?}};\\ \overline{\overline{F\widehat{x;y};\ *K_2;\ >\widehat{x;y}\text{ - ?}};\ *K_1;\ \in\widehat{y;N}\text{ - ?}}\end{array}\right), \tag{1}$$

где a и b – введенные значения переменных x и y; $*K_0$, $*K_1$ и $*K_2$ – сообщения о невыполнении условий; $F\widehat{x;y}$ – поиск наибольшего общего делителя вычислением остатка от деления x на

y, что запишем как $\%\overparen{x; y}$, где % – обозначение функционального унитерма поиска остатка от деления. Приписывание остатка переменной r имеет вид $=\overparen{r; \%\overparen{x; y}}$. Проверка значения остатка на неравность нулю имеет такое описание $\neq\overparen{r; 0}$. Если остаток не равен нулю, то переменной x приписывается значение переменной y ($=\overparen{x; y}$), а переменной y – значение остатка r ($=\overparen{y; r}$), и происходит возврат в цикл на вычисление нового остатка от деления. Если же остаток равен нулю, то делитель и есть искомым наибольшим общим делителем ($=\overparen{n; y}$). Этот процесс описывается формулой алгоритма

$$F\overparen{x; y} = \not\supset r \left(\begin{array}{l} =\overparen{r; \%\overparen{x; y}} \\ ; \\ \hline \left(\begin{array}{l} =\overparen{x; y}; =\overparen{n; y}; \neq\overparen{r; 0} - ? \\ ; \\ =\overparen{x; r} \end{array}\right. \end{array}\right. \qquad (2)$$

Подстановка формулы (2) в выражение (1) дает такую формулу

$$\left(\begin{array}{l} \left(\begin{array}{l} \leftarrow\overparen{x; b}; \\ \hline \leftarrow\overparen{y; b}; {}^*K_0; \in\overparen{x; N} - ?\ ; \end{array}\right. \\ \hline \not\supset r;\ {}^*K_2; >\overparen{x; y} - ?;\ {}^*K_1; \in\overparen{y; N} - ? \\ \left(\begin{array}{l} =\overparen{r; \%\overparen{x; y}} \\ ; \\ \hline \left(\begin{array}{l} =\overparen{x; y}; =\overparen{n; y}; \neq\overparen{r; 0} - ? \\ ; \\ =\overparen{x; r} \end{array}\right. \end{array}\right. \end{array}\right.$$

Полученной формулой опишем поиск наибольшего общего делителя для отдельных значений переменных x и y. Для того чтобы она описала поиск наибольшего общего делителя для всех натуральных чисел, ее индивидные переменные x и y необходимо связать операциями циклов. Индивидные переменные x и y свяжем операцией циклического секвенцирования, а индивидную переменную r – операцией циклического элиминирования, что даст такую формулу алгоритма:

$$\not\subset x \not\subset y \left(\begin{array}{l} \left(\begin{array}{l} \leftarrow\overparen{x; b}; \\ \hline \leftarrow\overparen{y; b}; {}^*K_0; \in\overparen{x; N} - ?\ ; \end{array}\right. \\ \hline \not\supset r;\ {}^*K_2; >\overparen{x; y} - ?;\ {}^*K_1; \in\overparen{y; N} - ? \\ \left(\begin{array}{l} =\overparen{r; \%\overparen{x; y}} \\ ; \\ \hline \left(\begin{array}{l} =\overparen{x; y}; =\overparen{n; y}; \neq\overparen{r; 0} - ? \\ ; \\ =\overparen{x; r} \end{array}\right. \end{array}\right. \end{array}\right. \qquad (3)$$

Теорема 2. Формула (3) описывает алгоритм выполнения условий и вычисление наибольшего общего делителя всех натуральных чисел.

Доказательство. Условиями есть принадлежность индивидных переменных x и y к натуральным числам, и то, что значение x больше значения y. Если не выполняется хотя бы одно из условий, например, $\in\overparen{x; N}$, то на основании теоремы 1 из формулы (1) получаем выражение, которое в дальнейшем преобразуется с учетом свойств операций секвенционных конъюнкций, дизъюнкции и инвертирования:

$$\left(\begin{array}{l} \leftarrow\overparen{x; a}; \\ \hline \vee\ \&\ \overparen{\in\overparen{x; N}, \left(\begin{array}{l} \leftarrow\overparen{y; b}, \&\ \overparen{\in\overparen{x; N}}; {}^*K_0, \\ \hline F\overparen{x; y}; {}^*K_2; >\overparen{x; y} - ?; {}^*K_1; \in\overparen{y; N} - ? \end{array}\right.} \end{array}\right. =$$

$$= \left(\begin{array}{l} \leftarrow\overparen{x; a}; \\ \hline \vee\ \&\ 0, \left(\begin{array}{l} \leftarrow\overparen{y; b}, \&\ 1; {}^*K_0, \\ \hline F\overparen{x; y}; {}^*K_2; >\overparen{x; y} - ?; {}^*K_1; \in\overparen{y; N} - ? \end{array}\right. \end{array}\right. =$$

$$= \left(\begin{array}{l} \leftarrow\overparen{x; a}; \\ \hline \vee\ 0,\ {}^*K_0, \end{array}\right. = \left(\begin{array}{l} \leftarrow\overparen{x; a}; \\ {}^*K_0. \end{array}\right.$$

Первая после знака равенства формула получается в результате замены условного секвенционного унитерма $\in\overparen{x; N}$ его значением – нулем, а его инвертирование дает единицу. Полученный алгоритм содержит присвоенное переменной x введенное значение и сообщение ${}^*K_0$ о том, что значение x не есть натуральным числом. Таким образом, алгоритмом есть невыполнение условия. Аналогично устанавливается невыполнение остальных условий.

Теперь рассмотрим случаи, когда условия выполняются. В таком случае из формулы (1) получим

$$\left(\begin{array}{l}\left(\begin{array}{l}\leftarrow \overset{\frown}{x;\, a};\\ \leftarrow \overset{\frown}{y;\, b}\end{array}\right.\\ F\,\overset{\frown}{x;\, y},\end{array}\right.$$

аналогично тому, как при невыполнении условий. Секвенцирование полученного выражения с формулой (2) – второй частью формулы (3), дает формулу, которая и есть алгоритмом поиска наибольшего общего делителя двух натуральных чисел. Доказательство второй части теоремы выполним на основе метода трансфинитной математической индукции [15, 16].

Сначала покажем, что формула (3) описывает поиск наибольшего общего делителя для начальных значений переменных x и y. Начальными значениями x и y есть два и 1. (десятичный знак). Подстановка этих значений в алгоритм (3) дает формулу

$$\left(\begin{array}{l}\left(\begin{array}{l}\leftarrow \overset{\frown}{x;\, 2}\\ \leftarrow \overset{\frown}{y;\, 1.}\end{array}\right.\\ \not\supset r\\ \left(\begin{array}{l}= \overset{\frown}{r;\, \%\, \overset{\frown}{2;\, 1.}}\\ \quad ;\\ \overline{\left(\begin{array}{l}= \overset{\frown}{x;\, y};\, = \overset{\frown}{n;\, 1.};\, \neq \overset{\frown}{0,\, 0}\text{ - ?}\\ \quad ;\\ = \overset{\frown}{x;\, r}\end{array}\right.}\end{array}\right.\end{array}\right. .$$

Очевидно, что унитерм $\neq \overset{\frown}{0, 0}$ принимает нулевое значение, а путем элиминирования по этим условиям получаем $= \overset{\frown}{n;\, 1.}$, что показывает приписывание переменной значение наибольшего общего делителя, равное единице. Следовательно, для начальных значений переменных формула (3) корректно описывает вычисление наибольшего общего делителя.

Аналогично устанавливается описание формулой (3) наибольшего общего делителя для произвольных значений переменной x и начального значения переменной y.

Выполним анализ алгоритма для i-го значения индивидной переменной x и j-го значения индивидной переменной y. Первая часть алгоритма (3) – формула (1), проанализированная ранее, и для данных значений индивидных переменных она не отличается. Относительно ее второй части – формулы (2), то после подстановки значений переменных для первой итерации получим

$$\left(\begin{array}{l}= \overset{\frown}{r_0;\, \%\, \overset{\frown}{i;\, j}}\\ \quad ;\\ \overline{\vee\left(\begin{array}{l}\& \neq \overset{\frown}{r;\, 0}\\ \quad ;\\ \left(\begin{array}{l}= \overset{\frown}{x;\, j}\\ \quad ;\\ = \overset{\frown}{y;\, r.}\end{array}\right.\end{array}\right. ,\ \overline{\left(\begin{array}{l}\& \neq \overset{\frown}{r;\, 0}\\ \quad ;\\ = \overset{\frown}{n;\, j}\end{array}\right.}}\end{array}\right.$$

Если значением функционального унитерма $\neq \overset{\frown}{r, 0}$ есть 0, то этот случай уже рассмотрен – наибольшим общим делителем есть j. Если же $\neq \overset{\frown}{r, 0} = 1$, то имеем выражение

$$\left(\begin{array}{l}= \overset{\frown}{r_0;\, \%\, \overset{\frown}{i;\, j}}\\ \quad ;\\ \left(\begin{array}{l}= \overset{\frown}{x;\, j}\\ \quad ;\\ = \overset{\frown}{y;\, r},\end{array}\right.\end{array}\right.$$

которое описывает приписывание переменной x значения j, а переменной y – значения r. Такое задание значений переменных связано с тем, что наибольший общий делитель следующей итерации не может быть большим остатка текущей итерации. Возвращение в цикл по операции циклического элиминирования приводит к формуле

$$\overline{\left(\begin{array}{l}= \overset{\frown}{r;\, \%\, \overset{\frown}{i;\, j}};\\ \quad ;\\ \left(\begin{array}{l}= \overset{\frown}{x;\, j}\\ \quad ;\\ = \overset{\frown}{y;\, r}\end{array}\right.\end{array}\right. \overline{\left(\begin{array}{l}= \overset{\frown}{r;\, \%\, \overset{\frown}{x;\, y}};\, \ldots;\, \neq \overset{\frown}{r;\, 0}\text{ - ?};\, \neq \overset{\frown}{r;\, 0}\text{ - ?}\\ \overline{\left(\begin{array}{l}= \overset{\frown}{x;\, y};\, = \overset{\frown}{n;\, y};\, \neq \overset{\frown}{r,\, 0}\text{ - ?}\\ \quad ;\\ = \overset{\frown}{y;\, r.}\end{array}\right.}\end{array}\right.}}$$

Этот процесс будет повторяться до тех пор (показано тремя точками), пока остаток от деления не будет равным нулю, а последнее уже рассмотрено. Тем самым формула описывает вычисление наибольшего общего делителя i-го

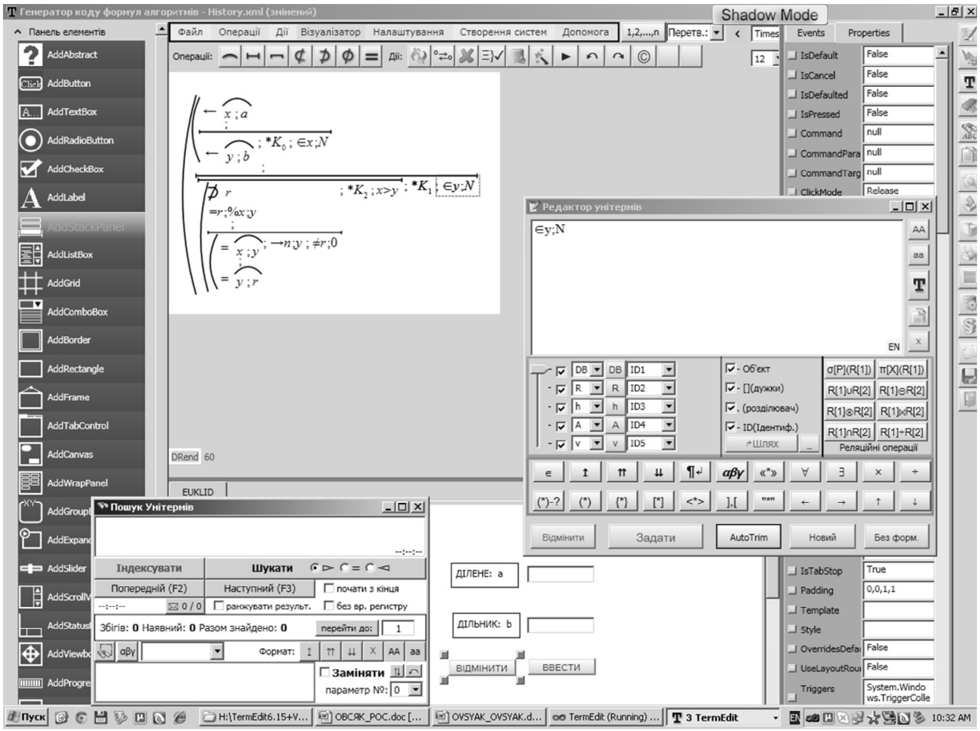

Рис. Главное окно инструментальных средств

значения переменной x и j-го значения переменной y.

Теперь необходимо доказать, что формула (3) описывает вычисление наибольшего общего делителя для всех значений переменных x и y. Для доказательства используем принцип совокупной математической индукции [16], что обусловлено наличием трех переменных. Третьей переменной есть r, для которой теорема доказана. Поскольку N – секвенционная область натуральных чисел ($N=\overset{\frown}{1,\overset{\frown}{2,\dots}}$), упорядоченная операцией секвенцирования, то относительно формулы (3) можно использовать *трансфинитную* математическую индукцию [16]. Обусловлено это тем, что доказательство для начальных и иных значений переменных x и y – аналогично. Для формулы (3) доказано, что она выполняется для произвольных значений i и j переменных x и y. Следовательно, на основании трансфинитной индукции, можно утверждать, что формула (3) описывает выполнение условий и вычисление наибольшего общего делителя всех натуральных чисел. Теорема доказана.

Инструментальные средства синтеза формул алгебры секвенционных алгоритмов

Система обозначений алгебры секвенционных алгоритмов – специфическая. Например, знак операции секвенцирования имеет форму эллипса, ширина (h) которого зависит от его длины (l) и вычисляется по формуле $h = =\sqrt{l}/+2$, у которой два – количество пикселов экрана монитора [17, 18], а зависимость ширины от длины знаков операций элиминирования и параллирования описываются выражением $h=\sqrt{l}/+2$ [16, 18]. Знаки операций секвенцирования, элиминирования и параллелирования имеют горизонтальную ($\frown$, $\vdash\!\dashv$, $\ulcorner\!\urcorner$) и вертикальную ((, $\top\!\!\!\bot$, [) ориентации. Геометрические размеры знаков зависят от геометрических размеров унитермов, которые упорядочиваются этими операциями. Для выбора и редактирования формул алгебры секвенционных алгоритмов могут быть использованы универсальные компьютерные системы: *Word*, *Coral Draw* и др. В связи с необходимостью выполнения большого количества действий, связанных с рисованием и масштабированием знаков операций, использование универсальных компьютерных систем для синтеза и редактирования формул алгебры секвенционных алгоритмов – малоэффективны. В связи с этим для синтеза формул алгебры секвенционных алгоритмов созданы специальные инструментальные средства.

Главное окно созданной системы компьютерного синтеза формул алгебры секвенционных алгоритмов показано на рисунке. В его верхней части обозначено название системы –

Генератор коду формул алгоритмів, меню команд с опциями: *Файл*, *Операції*, *Дії*, *Налаштування*, *Створення систем*, *Візуалізатор*. Непосредственно под меню команд – инструменты, предназначенные для автоматического создания знаков операций: секвенцирования, элиминирования, параллелирования, циклических секвенцирования, элиминирования, параллелирования и равенства. Выполнение таких действий, как замена унитермов, их реверсирования и элиминирование, задания групповых свойств, открытия доступа к базе алгоритмов, инициализации мастера создания баз данных из формул алгоритмов и перестройки упорядочений. Под инструментами расположено рабочее поле для синтеза и визуализации формул алгоритмов. На рабочем поле изображены синтезированные формулы алгоритмов. Ниже рабочего поля расположен рабочий стол для создания интерфейсов компьютерных систем. На рабочем столе показан тривиальный интерфейс компьютерной системы. В левой части окна расположена панель элементов с одним абстрактным графическим элементом (*AddAbstract*) и другие типичные графические унитермы (кнопка – *AddButton*, этикетка – *AddLabel*, элемент безальтернативного выбора – *AddRadioButton*, элемент альтернативного выбора – *AddChecBox* и др.), предназначенные для создания графических интерфейсов компьютерных систем. В правой части окна расположены списки свойств (*Properties*) и событий (*Events*) абстрактных и типовых графических унитермов.

В табл. 2 показана сравнительная оценка эффективности по количеству технологических операций, которые необходимо выполнить при синтезе алгоритма решения квадратного уравнения созданными инструментальными средствами и информационной технологией в сравнении с использованием универсальной (*Word*) и специализированных компьютерных систем Абстрактал [17] и Модал [18].

Т а б л и ц а 2. Сравнение эффективности инструментальных средств

Количество операций при наборе	Инструментальные средства и информационная технология						
	Созданные инструментальные средства и информационная технология	*Microsoft Word*		АбстрактАл		Модал	
		Количество операций	Уменьшено (раз)	Количество операций	Уменьшено (раз)	Количество операций	Уменьшено (раз)
Унитермов	60	110	1,80	75	1,20	99	1,65
Знаков операций секвенцирования	25	96	3,90	45	1,80	60	2,40
Знаков операций секвенцирования	12	30	2,50	36	3,00	40	3,30
Алгоритма в целом	97	236	**2,40**	156	**1,50**	199	**2,00**

На основании инструментальных средств разработаны системы компьютерной оптимизации формул алгебры секвенционных алгоритмов по количеству унитермов и генерирования реляционных баз данных по формулам алгебры секвенционных алгоритмов.

Заключение. Впервые в алгебру алгоритмов введено определение операции реверсирования с установлением связей между операциями секвенцирования, элиминирования и параллелирования, что обеспечило оптимизацию и наглядность, уменьшение затрат на реализацию формул алгоритмов. Впервые операция элиминирования описана формулой секвенционной логики над упорядоченными логическими переменными, которая может быть использована для доказательства непротиворечивости алгебры секвенционных алгоритмов.

Алгебра секвенционных алгоритмов первого порядка обеспечивает формализованное описание и обработку структур драных. Алгеброй секвенционных алгоритмов второго порядка формально описываются и обрабатываются алгоритмы, результат которых – компьютерный синтез и оптимизация, исследование и реализация алгоритмов с использованием инструментальных средств.

Созданы эффективные инструментальные средства и информационная технология синтеза, редактирования и оптимизации по количеству унитермов формул алгебры секвенционных алгоритмов.

Перспективны решения на основании алгебры секвенционных алгоритмов задач динамического программирования, компьютерного исследования достоверности информационных систем и генерирования программного кода из формул алгоритмов.

1. *Kleene S.C.* Origins of recursive function theory // Annals of the Theory of Computing. – Jan.1981. – N 1, **3**. – P. 52–67.
2. *Church A.* An unsolvable problem of elementary number theory // American J. of Mathematics. – 1936. – vol. 1, **58**. – P. 345–363.
3. *Turing A.M.* On computable numbers, with an application to the Entscheidungs problem // Proc. of London Mathematical Soc. – 1936–1937. – Ser. 2, **42**. – P. 230–265.
4. *Post E.L.* Finite Combinatory Processes – Formulation 1 // J. of Symbolic Logic. – 1936. – N 1. – P. 103–105.
5. *Марков А.А.* Теория алгорифмов // Тр. МИАН. – 1951. – Т. 38. – С. 176–189.
6. *Колмогоров А.Н.* О понятии алгоритма // УМН. – 1953. – Т. 8, **4** (56). – С. 175–176.
7. *Schönhage A.* Universelle Turing Speicherung / Automatentheorie und Formale Sprachen / J. Dörr, G. Hotz (Eds.) // Mannheim: Bibliogr. Institut, 1970. – P. 369–383.
8. *Aho A.V.*, *Hopcroft J.E.*, *Ullman J.D.* The design and analysis of computer algorithms. – Addison–Wesley Publ. Comp., 1974. – 470 p.
9. *Криницкий Н.А.* Алгоритмы вокруг нас. – М.: Наука, 1984. – 224 с.
10. *Цейтлин Г.Е.* Введение в алгоритмику. – К.: Сфера, 1998. – 310 с.
11. *Цейтлин Г.Е.*, *Захария Л.М.* Алгебраические аспекты полноты: абстракции, биология и экология // Проблеми програмування. – 2008. – № 2–3. – С. 31–36.
12. *Цейтлин Г.Е.*, *Мохница А.С.* Что такое алгебраическая алгоритмика? // Проблеми програмування. – 2009. – № 2–3. – С. 52–58.
13. *Овсяк В.К.* Программа имитации функционирования многозначных логических структур (УПОРЯДОЧЕНИЯ). № АП0159. – Киев: Укр. РФАП, 1987. – 450 с.
14. *Овсяк В.К.* Засоби еквівалентних перетворень алгоритмів інформаційно-технологічних систем // Доп. НАН України. – 1996. – № 9. – С. 83–89.
15. *Овсяк В.*, *Овсяк О.*, *Петрушка Ю.* Секвенційна логіка // Комп’ютерні технології друкарства. – 2011. – № 27. – С. 74–80.
16. *Математическая* энциклопедия / Гл. ред. И.М. Виноградов. – Т. 3. – М.: Советская энциклопедия, 1982. – 1183 с.
17. *Василюк А.С.* Підвищення ефективності математичного і програмного забезпечення редактора формул алгоритмів: Автореф. дис. … канд. тех. наук. – Львів, 2008. – 20 с.
18. *Бритковський В.М.* Моделювання редактора формул секвенційних алгоритмів: Автореф. дис. … канд. тех. наук. – Львів, 2003. – 18 с.

Поступила 12.05.2012

Тел. для справок: +38 032 223-6766, +38 097 109-3444, +38 097 110-1090 (Львов)
E-mail: *ovsyak@rambler.ru*, *ovsjak@ukr.net*

●