Oleksandr OVSYAK
NATIONAL UNIVERSITY OF CULTURE AND ARTS,
Shuchevitsha 5, 79-020 Lviv, Ukraine

Comparison of algebraic methods for algorithm transforms

Ph.D. eng. Oleksandr OVSYAK

He specializes in theoretical and applied computer science, theory of algorithms, programming, information systems, mathematical modeling of systems, computer simulation and mathematical modeling. He works as an associate professor in National University of Culture and Arts.

e-mail: ovsjak@ukr.net

Abstract

Methods of intuitive and algebraic description of algorithms are presented in the paper. Algebraic methods are compared in spite of their operation systems. Comparison of operations, their properties and classes of described algorithms are shown in the paper. Possibilities of expanding modern object programming languages based on the use of modified algebra algorithms are also shown. Comparing formulas of modified system of algorithmic algebras and algebra algorithms are presented and compared.

Keywords: algorithm algebra system, operation properties, algorithm formula, uniterm.

Porównanie metod algebraicznych przetwarzania algorytmów

Streszczenie

W artykule przeanalizowano metody intuicyjnego oraz algebraicznego opisu algorytmów. Obecnie istnieją cztery systemy algebr algorytmów: system algebr algorytmicznych Głuszkowa [4], modyfikacja tego sytemu wprowadzona przez Zeitlina [5], algebra algorytmów [6] oraz modyfikacja tej algebry, wprowadzona przez autora [7, 8]. Podkreślono zalety wykorzystania metod algebraicznych i ich przewagę nad metodami intuicyjnymi. Metody algebraiczne porównano pod względem używanych zbiorów operacji, właściwości operacji oraz klas opisywanych algorytmów. Pokazano że system algebr algorytmicznych Głuszkowa oraz jego modyfikacja wykorzystują operacje logiczne, takie jak koniunkcja i dysjunkcja, podczas gdy algebra algorytmów oraz zmodyfikowana algebra algorytmów wykorzystują operacje sekwencjonowania i zrównoleglenia. System algebr algorytmicznych oraz jego modyfikacja wykorzystują do uporządkowania operację kompozycji, która ma właściwość łączności. Podobnie jak w algebrze algorytmów tak i w jej modyfikacji do opisu kolejności wykorzystywana jest operacja sekwencjonowania, która ogólnie nie jest łącznościowa. Tym samym algebra algorytmów oraz jej modyfikacja rozszerzają możliwości opisywanych algorytmów, uwzględniając klasę algorytmów nie łącznościowych. Tej klasy nie uwzględnia system algebr algorytmicznych Głuszkowa oraz jego modyfikacja sformułowana przez Zeitlina. Pokazano możliwości rozszerzenia języków współczesnego programowania obiektowego przez wykorzystanie operacji algebry algorytmów oraz jej modyfikacji. Może to doprowadzić do uproszczenia zapisu algorytmów przy zachowaniu ich właściwości. Porównano właściwości trzech systemów algebraicznych: algebry algorytmów zmodyfikowanej przez autora, znanej algebry algorytmów, oraz zmodyfikowanego systemu algebr algorytmicznych Głuszkowa – Zeitlina. Podano wyniki porównania formuł algorytmów systemu algebr algorytmicznych i zmodyfikowanej przez autora algebry algorytmów. Przedstawiono możliwości uporządkowania zmiennych funkcji wielu argumentów.

Słowa kluczowe: algebra algorytmów, system algebr algorytmicznych, właściwości operacji, formuła algorytmu, uniterm.

1. Introduction

In current automation, measurement and control systems specialized information technologies are used, operating in full compliance with the algorithm implemented in hardware, software or in both.

There are known intuitive and algebraic methods for description of algorithms. Intuitive methods include verbal description, block diagram technique and methods of Post machine [1] Turing machine [2], Kolmogorov machine [3], Schönhage machine, Aho, Ullman and Hopcroft machine, lambda account, recursive functions, Markov algorithms, universal algorithm by Krinitski.

Methods of algebraic representation of algorithms are a system of algorithmic algebras [4], a modified system of algorithmic algebras [5], the algebra algorithms [6] and modified algebra algorithms [7, 8]. Algebraic methods show benefits and certain advantages over intuitive ones. These include in particular the possibility to solve the following tasks:

- Synthesis of mathematical models of algorithms;
- The model conversion in order to reduce the cost of implementation;
- Creating of data structure models;
- Primary model testing prior to its practical implementation and testing.

It is feasible to perform a comparative analysis of methods with the aim of the choosing of the best one designated to the specified application in automation, control or measurement system.

2. Comparison of algebraic methods of operation set

Table 1 shows the operation set of algorithmic algebras system and its modifications, as well as algorithm algebra and its modifications.

Tab. 1. Operations of algebraic methods
Tab. 1. Operacje metod algebraicznych

No	The system of algorithmic algebras and its modification		Algebra of algorithms and its modification	
	Operation	Designation	Operation	Designation
1.	Conjunction	$x\&y$	No	
2.	Disjunction	$X\vert Y$	No	
3.	Negation	$\overline{x}$	Reversal	$\overline{x}$
4.	Prediction	$X\bullet u$	No	
5.	Composition	$X*Y$	Sequencing	$\overset{\frown}{X;Y}$
6.	Alternative	$([u]\,X,\,Y)$	Elimination	$\overline{X;Y;u}$
7.	Cycle	$\{[u]\,X\}$	Cyclic sequencing	$\not\subset uX$
8.	No		Cyclic elimination	$\not\supset uX$
9.	No		Cyclic paralleling	$\emptyset uX$
10.	Filter	$F(u)$	No	
11.	Asynchronous	$X\vert\vert Y$	Paralleling	$\ulcorner X;Y \urcorner$
12.	Checkpoint	$T(u)$	No	
13.	Synchronizer	$S(u)$	No	

In algorithm algebra and its modifications there are no operations of conjunction, disjunction, prediction, filtering, checkpoint and synchronizing [6, 7, 8]. In algorithm algebra and its modification, operations of sequencing and paralleling on logical variables with the same index have the same properties as operations of conjunction and disjunction. Therefore there is no need for algorithm algebra and its modifications to introduce operations of conjunction and disjunction. The system of algorithm algebras by Glushkov [4] and its modifications by Zeitlin [5] have no operations of cyclic elimination and cyclic paralleling.

In algorithm algebra and its modification [6, 7, 8] there are available the cyclic elimination and cyclic paralleling operations, that do not exist in the systems of Glushkov and Zeitlin. It provides the possibility of compact representation of exclusive and parallel processes. Operation of cyclic elimination is replaced by multiple and successively applied elimination operation.

In modern languages of object-oriented programming, such as C#, elimination is implemented with the help of instruction *if*. Similarly, the sequencing operation is implemented by instruction of cycle *for*. The cycle elimination operation can be realized e.g. by instruction *switch*.

Similarly, as cyclic sequencing compactly describes the sequential iterative process, the operation of cycle paralleling describes parallel processes. In modern object-oriented program languages there is instruction cycle *for*, but there is no instruction that would implement paralleling cyclic operation.

Table 2 shows comparison of operation properties of algebraic methods.

As one can see from Table 2 modified algebra algorithms has elimination operation for multi meaningful condition and properties that describe connections between operations of sequencing, elimination, paralleling and reverse [8]. Operations of elimination on multi meaningful condition do not exist either in algebra of algorithms or in the system of algorithmic algebras and its modifications. Algebra of algorithms is missing connections between operations of sequencing, elimination, paralleling and reversing. The system of algorithmic algebras and its modifications is missing as well connections between the operations of negation, prediction, alternative, cycle, filter, asynchronous disjunction, checkpoint and synchronization.

Tab. 2. Comparison of operation properties of algebraic methods
Tab. 2. Porównanie właściwości operacji metod algebraicznych

Indicators	The system of algorithmic algebras and their modification	Algebra of algorithms	Modified algebra of algorithms
Indexing of components of operations	No	Yes	Yes
Operation of cyclic paralleling	No	Yes	Yes
Operation of cyclic elimination	No	Yes	Yes
Multi meaningful elimination	No	No	Yes
Connections between operations	No	No	Yes
Logical operations of conjun-ction and disjunction	Yes	No	Yes
Associativity of composition and sequencing operation	Yes	Yes for operations over components with the same index and no over components with different indices	Yes
Polymorphism	Yes	Yes	Yes
Description of associative algorithms	Yes	Yes	Yes
Description of non-associative algorithms	No	Yes	Yes

In the system of algorithmic algebras and its modification the composition operation describes the sequence. This operation is associative.

In algorithm algebra and its modification [6, 7, 8] the sequence is described by the sequencing operations. In case of variables with the same index the sequencing operation is associative. But when variables have different indices the sequencing operation is non-associative.

Thus a system of algorithmic algebras and its modification describes class of associative algorithms., while algebra of algorithms and modified algebra of algorithms describe both algorithm classes, those of associative and non-associative.

3. Comparison of the properties of operations

Table 3 shows the comparison of properties of operations of algorithm algebra and modified algorithm algebra.

Tab. 3. Comparison of the properties of the algorithm algebra and its modification
Tab. 3. Porównanie właściwości algebry algorytmów i jej modyfikacji

No	Operation, property	Algebra of algorithms	Modified algebra of algorithms
1	Multi meaningful elimination	No	Yes
2	Connection between sequencing and elimination	No	Yes
3	Connection between sequencing, paralleling and elimination	No	Yes
4	Connection between cyclic sequencing and paralleling	No	Yes
5	Connection between cyclic paralleling and sequencing	No	Yes
6	Connection between cyclic elimination and its reversal	No	Yes
7	Operations on logical constants	No	Yes
8	Sequential functional variables	No	Yes

As one can see from Table 3 modified algebra of algorithms has the following connections, between:
- operations of sequencing and elimination;
- sequencing, paralleling and elimination,
- cyclic sequencing and paralleling,
- cyclical paralleling and sequencing,
- cyclical elimination and its reversal,
- operations on logical constants (0 and 1) are available,
- sequential functional variables are available [8].

4. Comparison of formulas of methods of algorithm algebras

Comparison of the efficiency of using of algebraic methods is considered on the well-known Euclidean algorithm. Block diagram of the algorithm is shown in Fig. 1.

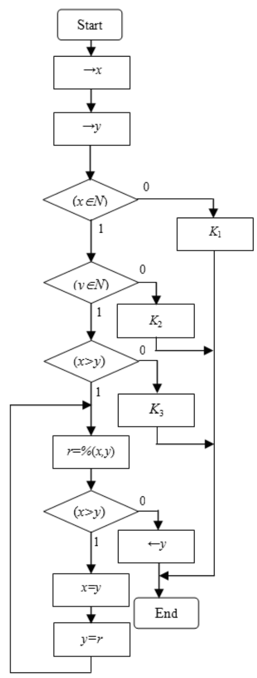

Fig. 1. Block diagram of the Euclidean algorithm
Rys. 1. Schemat blokowy algorytmu Euklidesa

The algorithm is designed for finding the greatest common divisor of two positive integers and has the following notation: $\rightarrow x$ - input of the first number, $\rightarrow y$ - input of the second number, N – set of natural numbers, $(x \in N)$ -? - check whether the entered value x is a natural number, $(y \in N)$ -? – check whether the entered value y is a natural number, messages: K_1 - x is not a natural number, K_2 - y is not a natural number, K_3 - x is less than or equal to y, $r = \%(x, y)$ - the remainder of dividing x by y and $\leftarrow y$ - output value of greatest common divisor.

After entering the x and y and control of conditions the remainder of dividing x by y is stored. If the remainder is zero (r=0), then the value of y is the greatest common divisor, and it is displayed in the screen ($\leftarrow y$). Otherwise, the variable y is attributed to the variable x, and the variable r- meaning remainder is attributed to the y variable, and r value is recalculated (cycle *for* $r\neq 0$).

Euclidean algorithm as a formula of modified system of algorithmic algebras is described as following

$$\to x * \to y * ([x\in N]\ ([y\in N]\ ([x>y]\ \{[\ (x\%\ y)\neq 0]\ x=y * y=x\%y\} * \leftarrow y,\ K_3),\ K_2),\ K_1)$$

and consists of 58 characters.

By modified algebra of algorithms the Euclidean algorithm is described by the following formula

$$\begin{array}{l} \left(\begin{array}{l}\to x\\ ;\\ \left(\begin{array}{l}\to y\\ ;\end{array}\right.\end{array}\right. \\ \hline \varnothing r;\ K_1;\ K_2;\ K_3;\ w \\ \left(\begin{array}{l} r=x\%y \\ ; \\ \hline \left(\begin{array}{l} x=y;\leftarrow y;r\neq 0 \\ ; \\ y=r\end{array}\right.\end{array}\right.\end{array} \qquad (1)$$

where

$$w=\overset{\frown}{x\in N;\ \overset{\frown}{y\in N;\ x>y}}. \qquad (2)$$

The number of characters in the formula is 56, so less than in the modified system of algorithm algebras. As one can see from the above formula expressions modified algorithm algebra are more visible than the formula of modified system of algorithmic algebras. Greater clarity of formulas of modified algebra of algorithms is achieved in case of graphical form of presentation of operation signs.

Considering the property of associativity of operations of composition of algorithmic algebra system and its modification [4, 5] the above formula of Euclidean algorithm can be written as follows

$$\to x * (\to y * ([x\in N]\ ([y\in N]\ ([x>y]\ \{[\ (x\%\ y)\neq 0]\ x=y * y=x\%y\} * \leftarrow y,\ K_3),\ K_2),\ K_1)).$$

In this case the following expression would be executed first

$$(\to y * ([x\in N]\ ([y\in N]\ ([x>y]\ \{[\ (x\%\ y)\neq 0]\ x=y * y=x\%y\} * \leftarrow y,\ K_3),\ K_2),\ K_1)),$$

and then the first number ($\to x$) would be introduced. Obviously, such sequence of algorithm gives the wrong result.

In this regard, the system of algorithmic algebras and its modification can be used only to describe the class of associative algorithms.

5. Sequential functional uniterms of modified algebra of algorithms

Functional uniterms, which variables are sorted be sequencing operation, are called sequential functional uniterms. For example, sequential functional uniterms are: addition, subtraction, multiplication and division

$$+\overset{\frown}{a,b},\ -\overset{\frown}{c,d},\ +\overset{\frown}{k,l},\ /\overset{\frown}{n:r}.$$

The use of sequencing operation in functional uniterms enables ordering of variables on which functional uniterms depend. For example, in a functional uniterm of division is important an order of variables that can be set using the sequencing operation.

6. Conclusions

1. By comparison of algebraic methods of algorithm representation for operation systems it was found that:
- a system of algorithmic algebras and its modification are bibasic, and algebra of algorithms and modification are monobasic systems;
- in algebra of algorithms and its modification there are no operations of conjunction and disjunction, filter, checkpoint and synchronizer, that exist in the system of algorithmic algebras and its modification. Operation functions of conjunction and disjunction in algebra of algorithms and its modification are substituted by operations of sequencing and paralleling;
- in a modified algebra of algorithms is available an operation of elimination at multimeaningful condition, there is no such operation in algebra of algorithms. The availability of operation of multimeaningful elimination reduces the number of elimination operations;
- the system of algorithmic algebras and its modification have no elimination operation at multimeaningful condition, cyclic elimination and cyclic paralleling.

2. By comparison of operation properties of algebraic methods it was found that:
- operations of sequencing and paralleling in algebra of algorithms and its modification assign indices defining the order of execution of uniterms. In the system of algorithmic algebras and its modification indices of execution order are not assigned;
- in algebra of algorithms and in the system of algorithmic algebras and its modification there are no connections between operations, while the modified algebra of algorithms has such connections. The connections provide the possibility to get algorithm formulas with fewer number of components;
- to describe the sequences in the system of algorithmic algebras and its modification, an operation of composition (multiplication), which is associative, is applied. The existence of properties of associativity in operation of composition limits the use of algorithmic algebra and its modification exclusively to describing the class of associative algorithms;
- by means of algebra of algorithms and its modification both classess of associative and non-associative algorithms can be described.

3. Introduction of sequential uniterms enables the ordering of component functional uniterms.

7. References

[1] Post E.L.: Finite Combinatory Processes – Formulation 1. Journal of Symbolic Logic, 1, pp. 103-105, 1936. Reprinted in The Undecidable, pp. 289ff.

[2] Turing A.M.: On computable numbers, with an application to the Entscheidungsproblem. Proceedings of London Mathematical Society, series 2, vol. 42 (1936-1937), pp. 230-265; correction, ibidem, vol. 43, pp. 544-546. Reprinted in [13 Davis M., pp. 155-222] and available online at http://www.abelard.org/turpap2/tp2-ie.asp

[3] Kolmogorov A.N.: On the concept of algorithm (in Russian). Uspekhi Mat. Nauk 8:4 (1953), pp. 175–176; translated into English in Uspensky V.A., Semenov A.L.: Algorithms: Main Ideas and Applications, Kluwer, 1993.

[4] Gluschkow W.M., Zeitlin G.E., Justchenko J.L.: Algebra. Sprachen. Programmierung. – Akademie-Verlag, Berlin 1980. – 340 p.

[5] Zeitlin G.E.: Algebraiczeskaja algorytmika: teoria i primenienija. Kibernetika i sistemnyj analiz. 2003, No 1, p. 8 – 18.

[6] Owsiak W., Owsiak A., Owsiak J.: Teoria algorytmów abstrakcyjnych i modelowanie matematyczne systemów informacyjnych. Wyd. Pol. Opolskiej, Opole, 2005.

[7] Owsiak W., Owsiak A.: Rozszerzenie algebry algorytmów. Pomiary Automatyka Kontrola. Vol. 56, 2010, No 2, s. 184-188.

[8] Ovsyak A.V., Ovsyak V.K.: Modificirowannaja algebra algorytmov i instrumentalnyje sredstva obrabotki formul algebry algorytmov. Upravlajuszczije systemy i maszyny. 2013, No 1, s. 27 – 36.

otrzymano / received: 04.07.2013
przyjęto do druku / accepted: 02.09.2013 *artykuł recenzowany / revised paper*