A CONCEPT OF A MODEL OF A COMPUTER SYSTEM FOR FORMING COMPLEX UNITERMS

Magdalena Niziołek, Volodymyr Ovsyak

***Abstract**: The existing methods of intuitive and mathematical methods of application of algorithms in system are described. It is shown that in the existing formula editor of algebra of algorithms only trivial (simple) uniterms are formed. In the work to secure the ability of creating complexes uniterms a concept of an adequate computer system model is created. The essence of this concept lies in the creation of function uniterms in the main class of the formula editor of algebra of algorithms and in specific sub-systems of complex uniterms, as well as in a modification of the XML described algorithms expressions generated by the editor.*

***Keywords.** Algebra, algorithm, model, formula, subsystem, uniterm: formatting, editor program.*

***ACM classification keywords**: F.2 Analysis of algorithms and problem complexity*

Introduction

Algorithms are usually presented in the form of a natural language description, a block-diagram or as a specialized language where the form of description is formalized for an unambiguity of the meaning. The most known methods for the last case are e.g. Post, Turing, Aho-Ullman-Hopcroft or Schönhage virtual machines, recursive functions (calculus λ, Church), Markov algorithms], Kolmogorov machine, as well as Krinitski universal algorithms.

Those methods, while easy to understand because of their intuitive notation, are not depicted formally. They contain notes of the sequence of operation needed to be effectuate to gain a certain result, but don't describe the relation between those sequences. Thus operations as optimization are difficult to perform.

The solution is to use formal description i.e. algorithm algebra [Owsiak, 2005]. Although in the formal description the algorithms aggravate their intuitiveness, the gain is in an abstraction that allows for certain transformations.

Definition of an uniterm

Uniterm is a concept that describe any given symbol or sign on which operation in algorithms algebra is executed. They can consist of numbers, letters or expressions e.g. 2, 34, -5, a, b, j,k, y>x, y=z^x, A(x), F(z,y,z).

Uniterms that don't depend on any variables can be divided into constants (e.g. 2,34, c, n), variables (x,y,z) and coefficients(i,j,k). Uniterms that depend on one or more variables are divided into concrete (e.g. $p=q+2$, $S_1=2x-3$) and abstract **(**e.g. $F(x)$, $R(x,y)$).

The existing formula editor of algebra of algorithms utilize only trivial (simple) uniterms (Img.1). However, it is easily to notice that in the case of variable depended uniterms the uniterms aren't a trivial expression, but are composed of other components. Those components can be algorithms or expressions that can be presented as formulas on their own.

As such a computer system that would take into account such complex uniterms needs to be created.

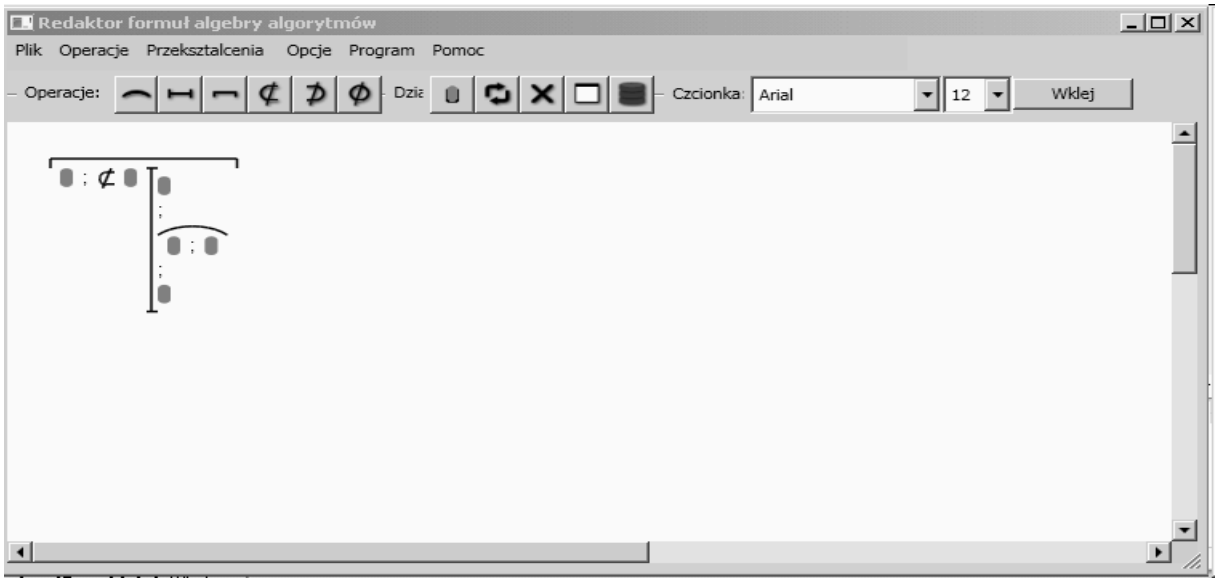

Fig. 1. Simple uniterms in the existent formula editor.

Formula editor for complex uniterms

The new system is an extension of the exiting one. In it's basic idea it's the same. However, the essence of this concept lies in the creation of complex (function) uniterms in the main class of the formula editor. Any uniterm can be changed into a complex uniterm, by creating a new object from UnitermComplex class (Fig. 2). The new complex uniterm is composed of unrestricted number of elements, though three is the most basic case (compare or assignment operations).

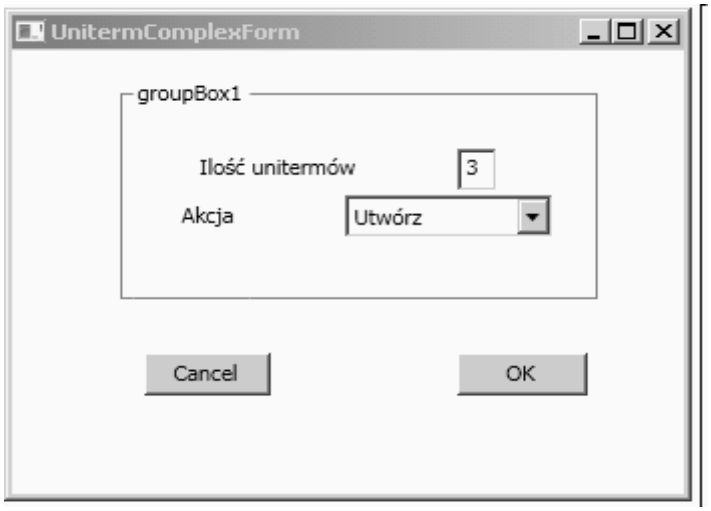

Fig. 2. Dialog for creating a new complex uniterm

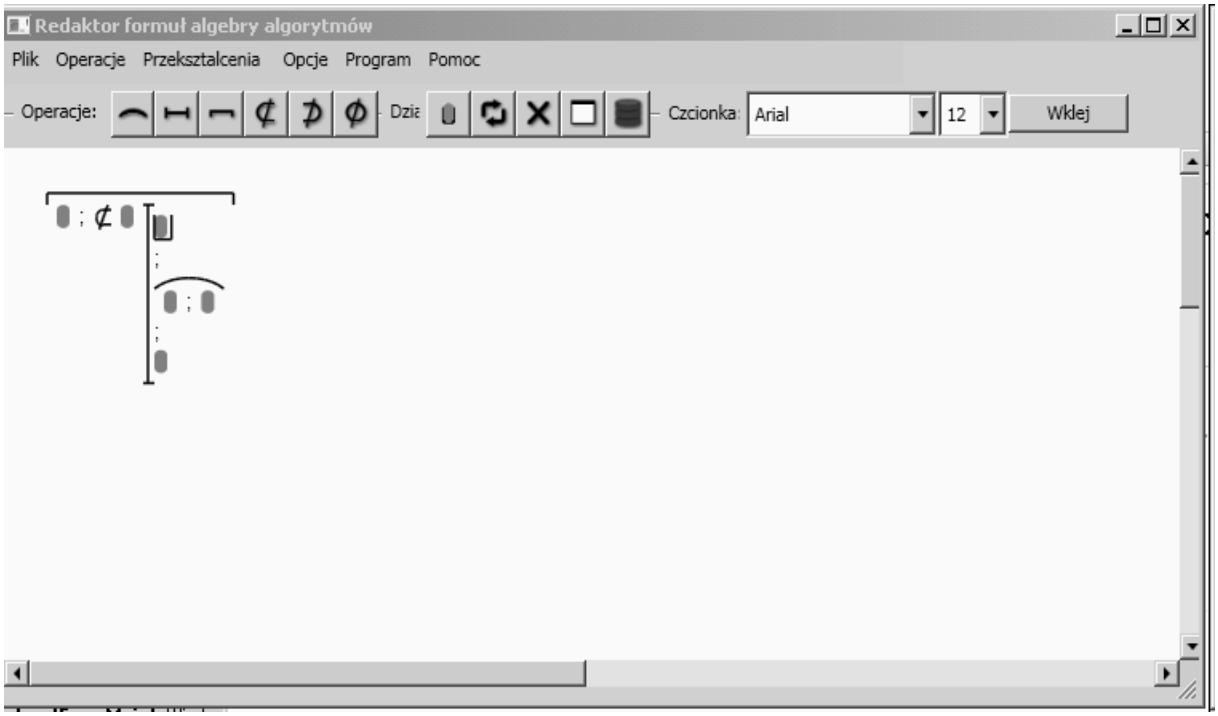

Fig. 3. Editor with one hidden complex uniterm (without data)

To differ from trivial uniterm the complex uniterm are symbolized by a container ⊔ . In the main form of the formula editor the complex uniterm can be shown in two ways: hidden – the complex uniterm is regarded as a trivial uniterm (Fig. 3), and visible – as an association of components (Fig. 4).

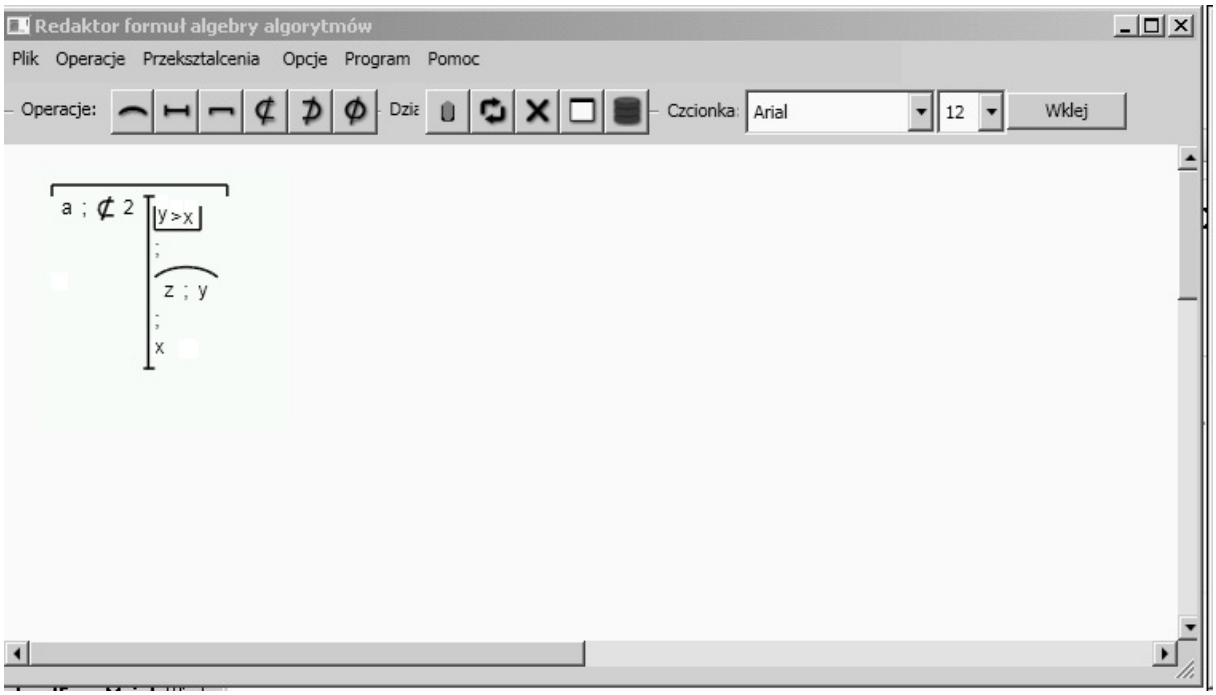

fig. 4. Editor with one visible complex uniterm (with data)

The original formula editor was written in Microsoft Visual Studio .NET 2010 with use of the modern languages XAML and C # . In the system the XML code, has hierarchical structure and access the individual elements is by the index. The index to node is text that contains more node names separated by slash marks. Each index starts from a node //root, which is superior to all other nodes, it means that it contains them. The uniterm node does not contain other under nodes. [Markuszewski, 2011].

In the new system an uniterm node can contain under nodes named subuniterm. Depending on the attribute the system will regard the subuniterm as a normal uniterm or as a new system.

Example of code:

```
public override void CreateXML(XmlDocument xmlDoc, XmlElement node)
    {
        XmlElement newElement;
        XmlAttribute newAttribute;

        newElement = xmlDoc.CreateElement("subuniterm");
        newAttribute = xmlDoc.CreateAttribute("action");
        if (action == Action.Create)
            newAttribute.Value = "Create";
        else if (action == Action.Visible)
            newAttribute.Value = "Visible";
        else
            newAttribute.Value = "Hidden";
        newElement.Attributes.Append(newAttribute);
        node.AppendChild(newElement);
        //if (condition != null) condition.CreateXML(xmlDoc, newElement);
        for (int i = 0; i < TermNrUC; i++)
        {
            if (termy[i] != null)
                termy[i].CreateXML(xmlDoc, newElement);
        }
    }
```

Conclusion

The original system regarded all uniterms as equal, regardless of their complex or stability (stationarity). The new system allows to expand the variable depending uniterms to a new subsystem, allowing a better accuracy and/or optimization of an algorithm written in algorithmic algebra.

Bibliography

[Markuszewski, 2011] T. Markuszewski. The model of the system for algorithm formula transformations, In: Measurement Automation and Monitoring, no 02/2011, pp. 201 – 204

[Owsiak, 2005] W.Owsiak, A.Owsiak, J.Owsiak. Teoria algorytmów abstrakcyjnych i modelowanie matematyczne systemów informacyjnych. – Opole: Politechnika Opolska, 2005. – 275 s.

Authors' Information

Magdalena Niziołek *– Opole University of Technology, Faculty of Electrical, Control and Computer Engineering, ul. Sosnkowskiego 5, 45-271 Opole, Poland; e-mail: m.niziolek@doktorant.po.opole.pl*

Major Fields of Scientific Research: programming, theory of algorithms,

Volodymyr Ovsyak *– Opole University of Technology, Opole, Poland and Ukrainian University of Printing, L'vov , Ukraine ; ovsyak@rambler.ru*

Major Fields of Scientific Research:Theoretical and applied computer science, theory of algorithms, programming, information systems, mathematical modeling.

THE COMPUTER PROCESS OF OPTIMIZATION ALGORITHMS FORMULAS

Tomasz Markuszewski

Abstract: *This paper presents a model of a computerized system to optimize formulas of algorithms. For this purpose the algebra of algorithms. The computer process optimization formula of algorithm is very complex, and were decomposed to simplify. The new model described in form of formulas algorithms, and contains variables, function uniterms. The variables are used for storing interim and final data used in the optimization process. Functional unierms are initiating variables, checking the possibility of optimization by introducing an additional condition, optimizing single operation algorithm algebra, and for all algorithms operations algebra with introducing an additional condition method. The built effect of the model is its simplicity. The primary benefit of implementation a model is to protect transformation formulas of algorithms.*

Keywords*: algebra, formula, algorithm, transformation, optimization, model, decomposition.*