

Model XML-description of compound uniterms of formulas algorithms

Volodymyr Ovsyak^{1,2}, Magdalena Niziolek², Julia Petruszka¹

¹Ukrainian University of Printing, UKRAINE, L'viv, Pidgolosko, street 19, E-mail: ovsjak@rambler.ru,

²Opole University of Technology, POLAND, Opole

The specialized computer editing program [1, 2, 3], are designed to automatize the processes of keing in and editing formulas algorithms. Algorithms are described as mathematical formulas by algebra of algorithms means [4]. Algorithms formulas are formed by trivial uniterms, trivial uniterms delimiters and operations signs. Trivial uniterms are text and graphic signs. Abstract graphical trivial uniterm is painted in gray rectangle with the size of 7x12 units. Text uniterms marks are formed by one or a number of text characters. Operations signs of algorithms algebra, in comparison with operations signs of classical algebra are complex graphic signs. Algorithms formulas formed by trivial uniterms and operations signs of algorithms algebra are complex mathematical expressions.

In this research, the reduce of algorithms formulas complexity is achieved due to the introduction of composite uniterms. Compound uniterms are those which are formed by two or more trivial uniterms.

To maintain the compound uniterms in the computer memory, the formula for generating of XML-description of compound uniterms, was created. The formula has a general access specifiers and two input uniterms – type of known subsystem variables [5, 6]. The formula introduces two uniterms-variables destined to store intermediate values. The formula uses functionality uniterms designed to create XML-description of compound uniterms and which are realized by well-known methods [5, 6]. The process of formation of a compound uniterm with m trivial uniterms is described by cyclic sequenation operation for variable i . For the variable i sequent value area is set.

Algorithm formula for XML-description of compound uniterms generation is programmatically realized by programming language C # of the platform Microsoft Visual Studio. NET.

Переклад виконано Малиновською О. А., центр іноземних мов «Universal Talk», www.utalk.com.ua

Модель XML-опису складених унітермів формул алгоритмів

Володимир Оасяк^{1,2}, Магдалена Нізьолек², Юлія Петрушка¹

¹Українська академія друкарства, УКРАЇНА, Львів, Підголоско, вул. 19, E-mail: ovsjak@rambler.ru,

²Опольський технологічний університет, ПОЛЬЩА, Ополье

Алгеброю алгоритмів описано формулу алгоритму комп'ютерного створення XML-опису складених унітермів формул алгоритмів.

Ключові слова – алгебра алгоритмів, модель, тривіальний унітерм, унітерм-змінна, функційний унітерм, програмний код.

I. Вступ

Відомими редакторами формул алгоритмів [1, 2, 3] створюються тривіальні унітерми. Використання виключно тривіальних унітермів суттєво збільшує складність формул алгоритмів. Для впорядкування великої кількості тривіальних унітермів необхідна і велика кількість операцій алгебри алгоритмів [4].

Унітерми є окремими текстовими знаками та їхніми послідовностями. А знаки операцій є графічними знаками. Для запису графічно-текстових формул алгоритмів у пам'ять комп'ютера необхідно перетворити їх у текстовий формат. З погляду наявності ефективних процедур створення та аналізу XML - формату є доцільним його використання для опису і нетривіальних унітермів.

II. Модель створення XML-опису складених унітермів формул алгоритмів

Тривіальний графічний абстрактний унітерм у комп'ютерній системі генерування програмного коду [3] є прямокутником шириною 7 і висотою 12 одиниць. Складеним унітермом називатимемо унітерм, який утворено двома і більше тривіальними унітермами. Формула (1) описує створення XML-опису складених унітермів формул алгоритмів. У формулі (1) застосовано такі умовні позначення: $cXML()$ – формула створення XML-опису складеного унітерму; pu – ідентифікатор відкритого доступу до функційних унітермів, який реалізується відомим [5, 6] ключовим словом `public` [5, 6]; $@dXML$, $@eXML$ і $@aXML$ – підсистеми, які реалізуються відомими класами `Xml Document`, `XmlElement` і `XmlAttribute` [5, 6]; $x\hat{I}@dXML$ і $y\hat{I}@eXML$ – вхідні для функційного унітерму $cXML()$ унітерми-змінні типу підсистем $@dXML$ і $@eXML$; $z\hat{I}@eXML$, і $t\hat{I}@aXML$ – створення унітермів-змінних z і t типу підсистем $@eXML$ і $@aXML$; $CreEl()$, $CreAtr()$, $Appe(i)$ $AppeChi()$ - функційні унітерми, які реалізовано відомими методами `CreateElement()`, `CreateAttribute()` і `Append()` [5, 6]; $z=x.CreEl("su")$ - приписування функційним

унітермом $CreEl()$ унітерму-змінній z значення унітерму-мінної x з дописаним значенням "su"; $t=x.CreAtr("ac")$ - приписування функційним унітермом $CreAtr()$ унітерму-змінній t значення унітерму-мінної x з дописаним значенням "ac"; ; $Val, AcCre, AcVis$ і $\$$ - унітерми, які реалізовано відомими властивостями Value, Action. Create, Action.Visible і константою null [5, 6]; $t.Val="Cre"$ - приписування функційним унітермом Val унітерму-змінній t значення "Cre"; $t.Val="Vis"$ - приписування функційним унітермом Val унітерму-змінній t значення "Vis"; $t.Val="Hid"$ - приписування функційним унітермом Val унітерму-змінній t значення "Hid"; $(a=AcCre)-?$ - перевірка наявності значення Cre унітерму-змінної a ; $(a=AcVis)-?$ - перевірка наявності значення Vis унітерму-змінної a ; $z.Attr.Appe(t)$ - приписування функційним унітермом $Attr.Appe()$ унітерму-змінній z значення унітерму-мінної t ; $y.AppeChi(z)$ - приписування функційним унітермом $AppeChi()$ унітерму-змінній y

значення унітерму-мінної z ; $\mathcal{A}(i \in \overline{0; 1; 2; \dots}) < m$ - цикл за змінною i , яка набуває значень із секвентної області

$$i \in \overline{0; 1; 2; \dots},$$

і ці значення є не більшими за m ; $u[i]$ - значення i -го тривіального унітерма; $u[i].cXML(x, z)$ - вибір функційного унітерму $cXML()$ з передачею йому значень унітермів-змінних x і z та попередньо створеного XML-опису $u[i]$. складеного унітерму.

$$\begin{aligned}
 & pu \ cXML(x \in @dXML, y \in @eXML) = \\
 & \left(\begin{array}{l}
 z \in @eXML \\
 t \in @dXML \\
 ; \\
 (z = x.CreEl("su"), \\
 t = x.CreAtr("ac")) \\
 ; \\
 t.Val = "Cre" \\
 ; \\
 t.Val = "Vis" \\
 ; \\
 t.Val = "Vis" \\
 ; \\
 (a = AcCre)-? \\
 ; \\
 (a = AcVis)-? \\
 ; \\
 z.Attr.Appe(t) \\
 ; \\
 y.AppeChi(z) \\
 ; \\
 (\mathcal{A}(i \in \overline{0; 1; 2; \dots}) < m) \\
 \left(\begin{array}{l}
 u[i].cXML(x, z); *; u[i] \neq \underline{\$} -?; *; (i < m) -? \\
 ; \\
 c_i
 \end{array} \right)
 \end{array} \right) \quad (1)
 \end{aligned}$$

III. Програмний код моделі

Програмний код моделі мовою C# має вигляд:
public override void CreateXML(Xml Document
xml Doc, Xml Element node)

```

{
    Xml Element newElement;
    Xml Attribute newAttribute;
    newElement =
        xml Doc. CreateElement("subuniterm");
    newAttribute =
        xml Doc. 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);
    for (int i = 0; i < TermNrUC; i++)
        { if (termy[i] != null)
            termy[i].CreateXML(xml Doc, newElement); }
}

```

Висновки

1. Формула алгоритму описує створення XML-опису складених унітермів.
2. Уведення складених унітермів зменшує складність формул алгоритмів.

Література

- [1] Бритковський В.М. Моделювання редактора формул секвенційних алгоритмів: автореф. дис. на здобуття наук. ступеня канд. тех. наук: спец. 01.05.02 "Математичне моделювання та обчислювальні методи" / Бритковський В.М. - Львів, 2003. - 18 с.
- [2] Василюк А.С. Підвищення ефективності математичного і програмного забезпечення редактора формул алгоритмів: автореф. дис. на здобуття наук. ступеня канд. тех. наук: спец. 01.05.02 "Математичне та програмне забезпечення обчислювальних машин і систем" / Василюк А.С. - Львів, 2008. - 20 с.
- [3] Овсяк О. Класи інформаційної системи генерування коду / О.Овсяк // Науковий журнал "Вісник Тернопільського державного технічного університету": "Тернопільський національний технічний університет імені Івана Пулюя". - № 1, 2010. - С. 171 - 176.
- [4] Owsiak W., Owsiak A. Rozszerzenie algebry algorytmów /Pomiary, automatyka, kontrola. № 2. 2010. - S.184-188.
- [5] Petzold C. Programowanie Windows w języku C#. - Warszawa: „RM”, 2002. - 1161 s.
- [6] Мэтью Мак-Дональд. Windows presentation foundation в .NET 3.5 с примерами на C# 2008. - Москва, Санкт-Петербург, Киев: "Apress", 2008. - 922 с.