

5. Балакришнан А.В. Теория фильтрации Калмана / Пер. с англ. – М: Мир, 1988. – С. 71–156. 6. Mohinder S. Grewal, Angus P. Andrews. Kalman Filtering: Theory and Practice using Matlab, Second Edition. – Wiley & Sons Interscience, 2001. – P. 114–165. 7. Лазарев Ю.Ф. Начала программирования в среде MatLAB: Учеб. пособие. – К.: НТУУ "КПИ", 2003. – С. 12–140.

УДК 004.4'232

В.К. Овсяк<sup>1,2</sup>, Т. Маркушевський<sup>2</sup>, Ю.В. Петрушка<sup>1</sup>

<sup>1</sup>Українська академія друкарства, Львів, Україна;

<sup>2</sup>Політехніка Опольська, Ополье, Польща

## МОДЕЛЬ ПІДСИСТЕМИ ВИЯВЛЕННЯ ШЛЯХІВ ДОСТУПУ ДО УНІТЕРМІВ

© Овсяк В.К., Маркушевський Т., Петрушка Ю.В., 2010

**Описано виконану декомпозицію підсистеми пошуку стежок доступу до унітермів формул алгоритмів, які подано спеціалізованим xml-подібним форматом. Підсистему утворено змінними і функційними унітермами. Побудовано засобами алгебри алгоритмів математичну модель підсистеми.**

**Ключові слова:** модель, функційний унітерм, змінні, декомпозиція, підсистема, xml-формат.

**We describe the decomposition done search engine, access to trails uniterms formulas algorithms that describe the special xml-similar format. Subsystem formed variables and featured uniterms. Algebra constructed by means of mathematical algorithms model subsystem.**

**Keywords:** model, featured uniterm, variables, decomposition, subsystem, xml format.

### Вступ

Алгоритми можна описати засобами інтуїтивних і формальних методів. Найпоширенішими методами інтуїтивного подання алгоритмів є вербальний [1] і блок-схемний [2]. Менш часто використовуються такі неформальні методи, як машин Поста [3], Тюрінга [4], Колмогорова [5], алгоритмів Маркова [6] та інші [7]. Формальними методами опису алгоритмів є алгебра алгоритмів [8] і розширена алгебра алгоритмів [9]. Система позначень операцій алгебри алгоритмів забезпечує наочне подання формул алгоритмів та має специфічні графічні знаки, яких немає серед стандартних математичних позначень. Набір і редагування формул алгоритмів можна виконати, наприклад, засобами таких універсальних комп'ютерних систем, як і іншими. Однак використання універсальних комп'ютерних систем для набору і редагування формул алгоритмів є малоефективним. Створені спеціалізовані редактори "Модал" [10], "Абстрактал" [11] і "Генкод" [12] істотно підвищують ефективність набору і редагування формул алгоритмів, але не автоматизують процесів мінімізації формул алгоритмів.

Формальне подання алгоритмів, порівняно з неформальним, поза іншими, має ту перевагу, що на підставі властивостей операцій алгебри алгоритмів можна автоматизувати процеси мінімізації формул алгоритмів.

Системою "Генкод" формули алгоритмів описано xml-подібним форматом. Щоб автоматизувати процеси мінімізації формул алгоритмів, необхідно встановити шляхи доступу до унітермів операцій формул алгоритмів, які описано у спеціальному xml-подібному форматі. У статті для розв'язання цієї задачі створено математичну модель підсистеми ідентифікації шляхів доступу до унітермів xml-формул алгоритмів. Засобами побудови математичної моделі підсистеми є алгебра алгоритмів і система позначень, уведена у дослідженнях [13, 14].

## 1. Декомпозиція підсистеми

### 1.1. Змінні

Для роботи з *xml*-файлом вводимо такі змінні:  $x$  типу *xmld* – документа, що запишемо як  $x\hat{I} @xmld$ , кількості унітермів ( $y\hat{I} @In$ ), на 1000 рядків одновимірну таблицю ( $t\hat{I} @Str(1000)$ ) для збереження встановлених шляхів доступу до унітермів.

### 1.2. Функційні унітерми

Введення адреси *xml*-файла описується функційним унітермом з назвою *Lo*, методом доступу *pu* і вхідним параметром  $s\hat{I} @Str$ , що запишемо як заголовок функційного унітерма  $pu Lo(s\hat{I} @Str)$ . Сам функційний унітерм містить задання початкового значення  $y=0$  та вибір функційного унітерма  $Lo(s)$ , тому

$$pu Lo(s\hat{I} @Str) = y=0; Lo(s).$$

Створюємо заголовок функційного унітерма введення *xml*-файла з одним вхідним параметром  $z\hat{I} @xmld$ , який має такий вигляд  $pu Lox(z\hat{I} @xmld)$  і зміст  $Lox(z\hat{I} @xmld) = (d=z)$ .

Функційний унітерм вибору рядків з *xml*-файла має заголовок  $pu Cr()$  і такий зміст

$$\overline{\dot{E}(i \leq d.Le); . ; (d.Le \neq 0)-?}$$

$$\left( \begin{array}{l} r_i = d.Cn \\ ; \\ Cl(r_i) \\ ; \\ c_i \end{array} \right)$$

де  $Le$  – стандартний унітерм обчислення кількості знаків;  $d.Le$  – обчислення кількості знаків значення змінної  $d$ ;  $Cn$  – стандартний унітерм вибору рядка з *xml*-файла;  $r_i = d.Cn$  – приписування змінній  $r_i$  вибраного рядка з *xml*-файла;  $Cl(r_i)$  – вибір функційного унітерма  $Cl()$  розпізнавання унітермів,  $c_i$  – ознака повернення у цикл, крапка у ролі другого унітерма операції елімінування є ознакою кінця функційного унітерма.

Функційний унітерм розпізнавання  $Cl()$  має заголовок  $pu Cl(k\hat{I} @xmln)$ , а виконувана ним функція описується такою формулою

$$\overline{\dot{E}(j \leq k.Le); . ; (k.Le \neq 0)-?}$$

$$\left( \begin{array}{l} q_j = k.Cn \\ ; \\ \overline{p = Ci(q_j); . ; (k.Na \neq "text")-?} \\ ; \\ c_j; . ; (j < k.Le)-? \\ Cl(q_j) \end{array} \right)$$

де  $xmln$  – стандартний тип даних *xml*-формату;  $Ci(q_j)$  – функційний унітерм встановлення стежки доступу до унітерма з таким описом

$$pu (w\hat{I} @Str)Ci(v\hat{I} @xmln) =$$

$$\left( \begin{array}{l} w = / + v.Na \\ ; \\ \overline{\dot{E}(v.Pn.Na \neq "document")-?} \\ \left( \begin{array}{l} w = / + v.Pn.Na + w; w = / + w; (v.Pn.Na \neq "document")-? \\ ; \\ v = v.Pn \\ ; \\ C(v.Pn.Na \neq "document") \\ ; \\ \overline{t_y = w; . ; ((w.Cn("uniterm")) \& v.Hcn) \neq true)-?} \\ ; \\ y = y + 1 \end{array} \right) \end{array} \right)$$



### 3. Фрагменти програмного коду моделі підсистеми

Модель підсистеми можна запрограмувати з використанням мов об'єктного програмування, наприклад, C++, Java, C#. Алгоритмічна мова C# є однією з найсучасніших мов об'єктного програмування, яка підтримується платформою Microsoft Visual Studio .NET [15] і середовищем Windows Presentation Foundation [16]. Мовою C# можуть бути запрограмовані консольні, віконні, баз даних і Web-сторінок комп'ютерні системи. C# є універсальною мовою об'єктного програмування і використовує вбудовану бібліотеку стандартних класів і методів, що істотно спрощує і полегшує написання програмного коду. У зв'язку з цим для написання коду вибираємо алгоритмічну мову C#.

#### 3.1. Код опису змінних

Змінну  $x$  типу *xmld* описуємо так:

```
public XmlDocument x;
```

де `public` – метод доступу до змінної; `XmlDocument` – тип даних; `x` – назва змінної та `;` – ідентифікатор кінця опису змінної. Код опису змінної

```
public int y;
```

Створення таблиці  $t$  з 1000 рядками є таким

```
public string[] t = new string[1000];
```

де `string` – тип даних таблиці; `[]` – ідентифікатор опису таблиці та `new` – оператор творення об'єкта.

#### 3.2. Програма функційного унітерма $ri\ Lo(s\hat{I}\ @xmld)$

У кодї заголовка ключове слово `void` означає відсутність видаваного процедурою значення, а перший рядок процедури описує створення об'єкта класу, у другому рядку задається значення змінної  $y$ , а третій рядок `x.Lo(s)` описує вибір методу завантаження *xml*-файла:

```
public void Lo(string s)
{
    x = new XmlDocument();
    y = 0;
    x.Lo(s);
}
```

#### 3.3 Код $Lox(z\hat{I}\ @xmld) = (d=z)$ :

```
public void Lo(XmlDocument z)
{
    d = z;
}
```

#### 3.4. Програма функційного унітерма $ri\ Cr()$

Для зчитування рядків *xml*-файла використовується інструкція `foreach`

```
public void Cr() {
    foreach (XmlNode r in d.ChildNodes) {
        Cl(r);
    } }
```

#### 3.5. Код функційного унітерма $ri\ Cl(k\hat{I}\ @xmld)$

Програма містить стандартні властивості `ChildNodes` та `Name`, ключове слово продовження виконання коду `continue`:

```
public void Cl(XmlNode k) {
    foreach (XmlNode q in k.ChildNodes) {
        if (q.Name == "#text") {
```

```

        continue;
    }
    string p = Ci(q);
    Cl (q);
}
}

```

### 3.6. Програма функційного унітерма $pu(w\hat{I}@Str)Ci(v\hat{I}@xmln)$

Код містить стандартні властивості Name, ParentNode та HasChildNodes і метод Contains():

```

private string Ci(XmlNode v){
    string w = "/" + v.Name;
    while (!(v.ParentNode.Name == "#document")){
        w = "/" + v.ParentNode.Name + w;
        v = v.ParentNode;
    }
    w = "/" + w;
    if (!w.Contains("uniterm") && w.HasChildNodes) {
        t[y]=w;
        y=y+1;
    }
    return w;
}

```

#### Підсумки

Декомпозицією підсистеми пошуку стежок доступу до унітермів формул алгоритмів, описаних *xml*-подібним форматом, на функційні унітерми зменшена складність проектування підсистеми.

Функційними унітермами описуються процеси задання адреси і вибір *xml*-файла формул алгоритмів, розпізнавання ключових слів і унітермів, формування і збереження шляхів доступу до унітерма.

Математична модель підсистеми однозначно описує процеси відшукування шляхів доступу до унітермів формул алгоритмів, поданих спеціалізованим *xml*-подібним форматом.

Програмою реалізацією та апробацією підтверджено коректність побудови і виконувани функції підсистеми пошуку шляхів доступу до унітермів.

1. Великий тлумачний словник сучасної української мови / Уклад. і голов. ред. В.Т. Бусл. – К.: Ірпінь: ВТФ «Перун», 2002. – 1440 с. 2. ЕСПД. Схемы алгоритмов программ, данных и систем. Условные обозначения и правила выполнения. ГОСТ 19.701 – 90 (ИСО 5807 – 85). – М.: Изд-во стандартов, 1991. – 27 с. 3. Post E.L. Finite Combinatory Processes – Formulation 1. *Journal of Symbolic Logic*, 1. – P. 103–105, 1936. 4. Turing A. M.: On computable numbers, with an application to the Entscheidungs problem. *Proceedings of London Mathematical Society, series 2, vol. 42 (1936–1937)*. – P. 230–265. 5. Колмогоров А.Н. О понятии алгоритма УМН. – Т. 8, вып. 4(56), 1953. – С. 175–176. 6. Марков А.А. Теория алгоритмов // *Тр. матем. ин-та АН СССР, им. В.А. Стеклова*, 38. – М.: Изд-во АН СССР, 1951. – С. 176–189. 7. Успенский В.А., Семенов А.Л. Теория алгоритмов: основные открытия и приложения. – М.: Наука, 1987. – 288 с. 8. Owsiak W., Owsiak A., Owsiak J. *Teoria algorytmów abstrakcyjnych i modelowanie matematyczne systemów informacyjnych*. – Opole: Politechnika opolska, 2005. – 275 s. 9. Owsiak W., Owsiak A. *Rozszerzenie algebry algorytmów / Pomiar, automatyka, kontrola*. – 2010. – № 2. – S. 184–188. 10. Бритковський В.М. Моделювання редактора формул секвенційних алгоритмів: автореф. дис. ...канд. тех. наук: спец. 01.05.02 “Математичне моделювання та обчислювальні методи” / В.М. Бритковський. – Львів, 2003. – 18 с. 11. Василюк А.С. Підвищення ефективності математичного і програмного забезпечення редактора формул алгоритмів: автореф. дис. ...канд. тех. наук: спец. 01.05.02 “Математичне та програмне

забезпечення обчислювальних машин і систем” / А.С. Василюк. – Львів, 2008. – 20 с. 12. Овсяк О. Класи інформаційної системи генерування коду / О. Овсяк // Вісн. Терноп. держ. техн. ун-ту: “Тернопільський національний технічний університет імені Івана Пулюя”. – 2010. – № 1. – С. 171–176. 13. Овсяк О.В. Граматика опису функційних унітермів / О.В. Овсяк // Поліграфія і видавнича справа // Української академії друкарства: Зб. наук. праць. – 2009. – № 2(50),– С. 18–22. 14. Овсяк О.В. Модель абстрактної підсистеми комп’ютерної інформаційної системи генерування коду (у цьому “Віснику”). 15. Petzold С. Programowanie Windows w języku C#. – Warszawa: RM, 2002. – 1161 s. 16. Мэтью Мак-Дональд. Windows presentejtation foundation в .NET 3.5 с примерами на C# 2008. – М., СПб., К.: Apress, 2008. – 922 с.

УДК 004.738.5

А.М. Пелецишин, Ю.О. Сєров, С.С. Федушко\*  
Національний університет “Львівська політехніка”,  
кафедри інформаційних систем та мереж,  
\*кафедра прикладної лінгвістики

## РОЗРОБЛЕННЯ АЛГОРИТМУ РЕЄСТРАЦІЇ ТА ВАЛІДАЦІЇ ПЕРСОНАЛЬНИХ ДАНИХ УЧАСНИКІВ ВЕБ-СПІЛЬНОТИ

© Пелецишин А.М., Сєров Ю.О., Федушко С.С., 2010

**Запропоновано методи перевірки персональних даних учасників веб-форуму. Ці методи реалізовані в алгоритмі реєстрації та валідації персональних даних про учасника веб-спільноти.**

**Ключові слова:** учасник веб-спільноти, персональний профіль, достовірність даних.

**This paper deals with the methods of verification of Web-forum members’ personal data, which are implemented in the algorithm of registration and validation of personal data on Web-community member.**

**Keywords:** Web-community member, personal profile, data verification

### Постановка проблеми

Розроблення нового підходу до верифікації даних, які надає веб-користувач під час реєстрації, є актуальним питанням в управлінні та модеруванні веб-спільнот. Зі збільшенням чисельності веб-спільнот та їх користувачів у мережі виникла потреба в розробленні методу перевірки максимальної кількості даних про потенційного учасника веб-форуму та комп’ютерно-лінгвістичного методу перевірки інформаційного сліду учасника інтернет-спільноти. За допомогою розроблення методів реєстрації учасників веб-спільноти спрощується реєстрація потенційних користувачів.

Для спрощення управління та підвищення ефективності діяльності веб-спільноти ми розробили основу для комп’ютерно-лінгвістичного методу перевірки інформаційного сліду учасника веб-форуму, який ґрунтується на лінгвістичному аналізі інформаційного наповнення Веб-спільноти. За допомогою комп’ютерно-лінгвістичного аналізу адміністрація спільноти зможе позбутись небажаних учасників та покращити комунікативну поведінку користувачів веб-спільноти.

### Аналіз останніх досліджень

Веб все більше набуває ознак реальної глобальної спільноти людей, у якій неможливо обійтись без механізмів ефективною ідентифікації [2]; відкритої системи формування і моніторингу комунікативної поведінки учасника глобальної спільноти тощо. Як наслідок, все актуальнішою стає перевірка достовірності інформаційного наповнення Вебу та інформації веб-особистості – множини