

О.В. Овсяк

Українська академія друкарства

МАТЕМАТИЧНА МОДЕЛЬ ПРОЦЕСУ АНАЛІЗУ *xml* – ФОРМАТУ ФОРМУЛ АЛГОРИТМІВ

*Засобами алгебри алгоритмів описана створена математична модель процесу аналізу *xml* – формату формул алгоритмів. Модель описує виявлення помилок в *xml* – описі формул алгоритмів, розпізнавання типу, орієнтації та розділювача унітермів операцій.*

*Algebra algorithms described by means of a mathematical model of analysis *xml* - format formula algorithms. The model describes the detection of errors in *xml* - formulas describing algorithms, pattern type, orientation, and separator uniterms operations.*

Вступ і формулювання задачі

Сучасні інформаційні системи є складними системами. Побудова моделей і реалізація моделей таких систем відбувається з використанням метода декомпозиції. Розбиттям системи на підсистеми зменшується складність як створення моделі так і її реалізації. Наприклад, модель сучасного об'єктного програмування базується на декомпозиції створюваної інформаційної системи на класи. Класи теж декомпонуються на складові, якими, наприклад, є методи (процедури), інтерфейси, делегати, перерахунки, властивості, поля, змінні, оператори, інструкції тощо [4, 5]. Складові класи та самі класи в алгебрі алгоритмів [1, 2, 3] названо унітермами. Методи (процедури) мов об'єктного програмування відповідають функційним унітермам алгебри алгоритмів [6, 7]. Класи комп'ютерної системи генерування програмного коду наведена у дослідженні [8]. Абстрактний клас *Term* призначений для задання процедур опрацювання унітермів. Формули алгоритмів у пам'ять комп'ютера записуються у вигляді *xml* – подібного формату даних [9]. Для виводу на екран комп'ютера зчитаних із пам'яті формул необхідним є виконання аналізу *xml* – подібного файлу з метою пошуку і вибору формульних даних. Ними є типи операцій, орієнтація (горизонтальна чи вертикальна), розділювач унітермів (кома чи крапка з комою) і самі унітерми. У роботі подано математичну модель функційного унітерма яка описує ці процеси аналізу.

Модель функційного унітерма аналізу XML-формули

Алгоритм аналізу утворимо *заголовком* і *змістом* (наповненням, суттю) алгоритму. Заголовок алгоритму утворено методом доступу, властивостями, вхідними і вихідними параметрами та назвою. Задаємо для алгоритму аналізу *xml* – файла загальний метод доступу (позначимо його літерами *pu*), статичну властивість (*st*), вхідні параметри, якими є змінна типу підсистеми *Term*, що запишемо як $t \in @T$ та змінна *n* типу стандартної підсистеми $@Nod$ (повною назвою якої є XmlNode [1, 2]), використовувана для вибору унітермів з *xml* – файла. Нехай вихідним є параметр $w \in @T$. Тому заголовок функційного унітерма має такий опис

$$pu\ st\ (w \in @T)\ Txml(t \in @T, n \in @Nod).$$

Формулою (1) задаємо функційний унітерм, який має такий зміст:

вираз $u = @U()$ описує створення об'єкта *u* класу *U* (скорочена назва класу *Uniterm*), де – ідентифікатор класу; $u.par = t$ – приписування змінній *par* класу *U* значення вхідної змінної *t*; $n.IT.Le$ – обчислення кількості символів вхідної змінної *n* з використанням стандартних процедур *IT* і *Le* (повні назви InnerText і Length [1, 2]); $u.val = n.IT$ – присвоєння змінній *val* класу *U* значення унітерма $n.IT$, яке вибирається стандартною властивістю *IT* із вхідної змінної *n*; $w = u$. – вихідній змінній надається значення об'єкта *u* і на цьому завершення виконання алгоритму (ідентифікатор крапка); $n.Na.Eq("uniterm")$ – порівняння значення вхідної змінної *n* і ключового слова *uniterm* з використанням стандартних властивості виділення назви *Na* і порівняння *Eq* (повні назви яких Name і Equals()); $z_1 = ori$; $z_2 = Ori$

$Q_0 = s; e; p; cs; ce; cp; \quad b_i \in Q_1 = S; E; P; CS; CE; CP; \quad j, k \in Q_2 = 0; 1;$

$Q_3 = s; p; \quad x_j \in Q_4 = sep; ori; \quad y_j \in Q_5 = Sep; Ori; \quad a_{0,k} \in Q_6 = Semi; Com;$

$A_{1,k} \in Q_7 = Hor; Ver; \quad t_i = \begin{cases} tA, \text{ якщо } i \in Q_8 = s, e, p, \\ cond, \text{ якщо } i \in Q_9 = cs, ce, cp; \end{cases} \quad r_i = \begin{cases} tB, \text{ якщо } i \in Q_8, \\ t, \text{ якщо } i \in Q_9; \end{cases}$

$\left(\begin{array}{l} u = @U() \\ ; \\ u.par = t \\ ; \\ \hline u.val = n.IT; w = u; (n.IT.Le > 0) - ?; \nexists (i \in Q_0); (n.Na.Eq("uniterm")) - ? \end{array} \right)$

$\left(\begin{array}{l} i = @b_i; Exce(„Пом. _A”); (i \in Q_0) - ? \\ ; \\ i.par = t \\ ; \\ \nexists j \nexists k \\ \left(\begin{array}{l} \overline{i.x_j = @b_i.y_j.a_{1,k}; c_k} \\ ; \\ Exce(„Помилка типу”) \\ ; \\ n.Attri[“x_j”].Val.Eq(“a_{j,k}”) - ? \\ ; \\ c_j \\ ; \\ \nexists k \\ \overline{i.z_1 = @b_i.z_2.a_{1,k}; c_k} \quad (1) \\ ; \\ Exce(„Помилка типу”) \\ ; \\ n.Attri[“x_1”].Val.Eq(“a_{1,k}”) - ? \\ ; \\ (i \in Q_3) - ? \\ ; \\ i.t_i = TF(i, n.CN[0]) \\ ; \\ i.r_i = TF(i, n.CN[1]) \\ ; \\ \hline i.cond = TF(i, n.CN[2]); *; (i=e) - ? \\ ; \\ w = i \\ ; \\ c_i \end{array} \right) \end{array} \right)$

Фрагмент програмної реалізації моделі

Однією з найсучасніших мов програмування є C#. Фрагмент коду моделі (1) має такий вигляд:

```
Uniterm u = new Uniterm();
    u.par = term;
    if (n.InnerText.Length > 0) u.val = n.InnerText;
    return u;
}
else if (node.Name.Equals("seq"))
{
    Sequence s = new Sequence();
    s.par = term;
    if
(n.Attributes["sep"].Value.Equals("semi"))
        s.sep = Sequence.Separator.Semicolon;
else if (n.Attributes["sep"].Value.Equals("com"))
        s.sep =
Sequence.Separator.Comma;
        else throw new
Екхепшн("Неправильний формат файла XML.");

if(n.Attributes["ori"].Value.Equals("hor"))
s.ori = Sequence.Orientation.Horizontal;
    else if
(n.Attributes["ori"].Value.Equals("ver"))
s.ori = Sequence.Orientation.Vertical;
        else throw new
Екхепшн("Неправильний формат файла XML.");
s.tA = TermFromXML(s, n.ChildNodes[0]);
s.tB = TermFromXML(s, n.ChildNodes[1]);
    return s;}
.
.
.

else if (n.Name.Equals("cp")) {
    CyclicParallelisation cp = new
CyclicParallelisation();
    cp.par = term;
    if
(n.Attributes["ori"].Value.Equals("hor"))
        cp.orientation =
CyclicParallelisation.Orientation.Horizontal;
    else if
(n.Attributes["ori"].Value.Equals("ver"))
        cp.orientation =
CyclicParallelisation.Orientation.Vertical;
    else throw new Екхепшн("Неправильний формат
файла XML.");
    cp.condition = TermFromXML(cp, n.ChildNodes[0]);
    cp.term = TermFromXML(cp, n.ChildNodes[1]);
    return cp;
```

```

    }
    else
    {
        throw new Exception("Неправильний формат файла
XML.");
    }
}

```

Висновки

1. Математична модель у компактному вигляді описує процеси аналізу *xml* – формату формул алгоритмів. Модель описує виявлення помилок в *xml* – описі формул алгоритмів, розпізнавання типу, орієнтації та розділювача унітермів операцій.

2. Формула алгоритму може бути використана для створення комп'ютерних інструментальних засобів динамічного програмування.

1. *Petzold C. Programowanie Windows w języku C#. – Warszawa: „RM”, 2002. – 1161 s.*

2. *Мэтью Мак-Дональд. Windows presentation foundation в .NET 3.5 с примерами на С# 2008. – Москва, Санкт-Петербург, Киев: “Apress”, 2008. – 922 с.*

3. *Owsiak W., Owsiak A., Owsiak J. Teoria algorytmów abstrakcyjnych i modelowanie matematyczne systemów informacyjnych. – Opole: Politechnika opolska, 2005. – 275 s.*

4. *Ovsiak V.K.: Computation Models and Algebra of Algorithms. http://www.nbuu.gov.ua/Portal/natural/VNULP/ISM/2008_621/01.pdf*

5. *Owsiak W., Owsiak A. Rozszerzenie algebry algorytmów / Pomiarы, automatyka, kontrola. – № 2, 2010. – S. 184 – 188.*

6. *Овсяк О. Моделі рекурсії та рекуренції / О.Овсяк // Вісник Національного університету “Львівська політехніка”: Комп'ютерні науки та інформаційні технології. – № 663, 2010. – С. 116 – 122.*

7. *Овсяк О.В. Граматика опису функційних унітермів / О.В.Овсяк // Поліграфія і видавнича справа: Збірник наукових праць Української академії друкарства. – № 2(50), 2009. – С. 18 – 22.*

8. *Овсяк О. Класи інформаційної системи генерування коду / О.Овсяк // Вісник Тернопільського державного технічного університету: “Тернопільський національний технічний університет імені Івана Пулюя”. – № 1, 2010. – С. 171 – 176.*

9. *Овсяк О. Модель формату текстового опису формул алгоритмів / Овсяк О. // Тези доповідей V міжнародної науково-практичної конференції “Сучасні проблеми і досягнення в галузі радіотехніки, телекомунікацій та інформаційних технологій”. - Запоріжжя, 22 – 24 вересня 2010 р. – С.185-187.*