ливо $g \approx m^2/2$ и $m \approx \sqrt{2g}$. Подставляя последний результат m в выражение для числа точек, получим

$$N_g(q) = 2\sqrt{2g} + 4(1 + \langle (q-1)/(4g^2) - 2 \rangle_3)g + 6g.$$

Граница Хассе – Вейля для максимальных кривых равна $N_g(q)_{HV} = q + 1 + 2g\sqrt{q}$. Вычисление $\limsup_{g \to \infty} N_g(q)/N_g(q)_{HV}$ даст (6). ◊

Точные вычисления $N_g(q)/g$ согласуются с результатами (5) и (6). Так, для $q = 257$ и $m = 16$ имеем $N = 816$ и $N_g(q)/g = 7{,}76$, а для $q = 2^{16} + 1 = 65537$ и $m = 2048$ имеем $N = 12589056$ и $N_g(q)/g = 6{,}01$.

ВЫВОДЫ

1. В простом поле не существует максимальных кривых Ферма. При большом роде проигрыш границе Хассе – Вейля пропорционален $1/(\sqrt{q})$. С уменьшением рода кривой значение числа точек приближается к границе Хассе – Вейля, и при $g = 0{,}1$ имеем тривиальный случай $N = q + 1$.

2. Универсальное хеширование по кривым Ферма в простом поле не обеспечивает требования по вероятности коллизии. В соответствии с выражением (2) для є алгебраическая кривая должна быть большого рода, с большим числом точек и значительным отношением $N_g(q)/g$. Асимптотика $N_g(q)/g$ для кривых Ферма равна 10. Кривые малого рода проигрывают по параметру N, что не обеспечивает требования по вероятности коллизии для практических значений q.

3. Результаты для кривых Ферма представлены для простого поля и частично могут быть отнесены к свойствам кривых в расширениях конечного поля. Расширенные конечные поля имеют большее многообразие по комбинаторным свойствам, что влияет на оценки параметров кривых.

СПИСОК ЛИТЕРАТУРЫ

1. *Bierbrauer, J.* On families of hash functions via geometric codes and concatenation. / Bierbrauer J., Johansson T., Kabatianskii G., Smeets B. // Advances in Cryptology-CRYPTO'93 Proceedings. – Springer-Verlag, 1994. – P. 331–342.
2. *Халимов, Г. З.* Коллизионные оценки универсального хеширования на основе схем с алгебраическими кодами / Халимов Г. З. // Прикладная радиоэлектроника. – 2009. – Т. 8, Вып. 3. – С. 338–342.

Надійшла 25.10.2010

Халімов Г. З.

ОЦІНКА ПАРАМЕТРІВ КРИВИХ ФЕРМА ДЛЯ УНІВЕРСАЛЬНОГО ГЕШУВАННЯ

Отримано точні рішення для числа точок кривих Ферма, коли порядок поля має дільники 2, 3 та 6, також оцінки числа точок на основі ймовірнісного підходу. Приводяться асимптотичні границі відношення максимального числа точок кривої Ферма в простому полі до її роду та до границі Хассе – Вейля.

Ключові слова: універсальне гешування, криві Ферма.

Khalimov G. Z.

ESTIMATION OF FERMA CURVES PARAMETERS FOR UNIVERSAL HASHING OF NUMBER SOLUTION FOR HURVITZ EQUATION IN THE FINITE FIELD

Exact solutions for a number of Ferma curves points when the field order has dividers 2, 3 and 6 as well as estimates of points number have been obtained using probabilistic approach. Asymptotic boundaries of the ratio of maximum number of Ferma curve points in a simple field to its genus and to Hasse–Weil boundary are given.

Key words: universal hashing, Ferma curves.

УДК 004

Овсяк О. В.

Канд. техн. наук, доцент Львівської філії Київського національного унів ерситету культури і мистецтв

МОДЕЛЬ РОЗШИРЕНОЇ НОТАЦІЇ ТЕКСТОВОГО ОПИСУ ФОРМУЛ АЛГОРИТМІВ

З метою запису у пам'ять комп'ютера графічно-текстових формул алгоритмів, які утворені фукційними унітермами з впорядкованими змінними і параметрами, створено розширену **xml**-подібну нотацію опису формул алгоритмів. Модель розширеної нотації описана засобами алгебри алгоритмів з використанням операції секвентування. Наведено приклад використання розширеної нотації для опису формули алгоритму Евкліда.

Ключові слова: модель, нотація, синтаксис, семантика, унітерм, секвенція, елімінування, паралелення.

ВСТУП

Для опису алгоритмів інформаційних технологій і систем найчастіше використовуються вербальний і блок-схемний методи. Відомо, що ці методи, як і методи машин Тюрінга [1], Поста [2], Колмогорова [3], Ахо – Ульмана – Хопкрофта [4], Шонхаге [5] і рекур-

сивних функцій [6], алгоритмів Маркова [7] та універсальних алгоритмів Кріниць кого [8] є методами неформального опису алгоритмів [9]. З метою математичного опису алгоритмів інформаційних технологій і систем створена алгебра алгоритмів [10] та її розширення [11]. Операції алгебри алгоритмів є оригінальними і мають специфічні позначення, яких немає серед стандартних математичних позначень. Тому для автоматизації процесів набору і редагування формул алгоритмів потрібно створити спеціалізовану комп'ютерну систему.

Як правило, кожна із комп'ютерних систем має свій формат даних, наприклад, **Word** використовує формат даних з розширенням **doc**. Останнім часом одним з найбільш часто використовуваних форматів є **xml** формат. На основі мови опису даних **XML** [12] створена сучасна мова **XAML** [13], призначена для програмування інтерфейсів прикладних інформаційних і комп'ютерних систем та комп'ютерної графіки. Нотація (система умовних письмових позначень [14]) **XAML**, у порівнянні з сучасною мовою об'єктного програмування **C#** [12], у більшій мірі зорієнтована на розробників комп'ютерної графіки.

ФОРМУЛЮВАННЯ ЗАДАЧІ

Спеціалізованими редакторами для набору і редагування формул алгоритмів є комп'ютерні системи **МОДАЛ** [15], **АБСТРАКТАЛ** [16] і **GenCod** [17]. Системи **МОДАЛ** і **АБСТРАКТАЛ** хоч і відрізняються своїми можливостями, але мають подібні формати даних. Подібність форматів даних полягає в наявності полів, які призначені для зберігання координат розташування графічних знаків операцій на робочому полі системи. Недоліком такого розв'язання є жорстка фіксація формул алгоритмів до місця розташування на робочому полі комп'ютерної системи. Для усунення цього недоліку у системі **GenCod** створено **xml-** подібний формат даних збереження формул алгоритмів у пам'яті комп'ютера. Однак формат **GenCod** має такий недолік, як відсутність можливості впорядкування змінних функційних унітермів. З метою усунення цього недоліку у статті введено модифікований формат текстового опису формул алгоритмів.

МОДЕЛЬ РОЗШИРЕНОЇ НОТАЦІЇ ОПИСУ ФОРМУЛ АЛГОРИТМІВ

Опис моделі розширеної нотації виконаємо засобами алгебри алгоритмів. Загальний вигляд текстового опису формули алгоритму є таким:

$$\overbrace{L\,;\ R_1\,;Q\ ;R_2}\ ,$$

де

$$L = \overbrace{<?xml;\ \#\!:\!v\ ;\#\ ;z;\#;\ ?>}\ ,$$

де *<?xml* – константа, ідентифікуюча початок опису заголовку формули алгоритму, *#* – наявність одного або декількох пропусків (пробілів), *v* – версія формату, *z* – формат кодування, *?>* – константа, ідентифікуюча кінець опису заголовку формули алгоритму,

$$v \in B = \overbrace{version="1.0",\ version="1.1",\ \ldots,\ version="X.Y"},$$

$$z \in Z = \overbrace{encoding="utf8",\ encoding="utf16"}\ ,$$

$$R_1 = \overbrace{<\ ;\ root\ ;\ >}\ ,$$

< – константа-ідентифікатор початку опису дескриптора, *root* – ідентифікатор кореневого дескриптора, *>* – ідентифікатор закриття початку і кінця опису дескриптора,

$$R_2 = \overbrace{</\ ;\ root\ ;\ >}\ ,$$

де *</* – константа-ідентифікатор опису кінця дескриптора,

$$Q = \overbrace{W_0\ ;\ W_1\ ;\ \ldots\ ;\ W_{n-1}},\ Q = \overbrace{W_0\ ;\ W_1\ ;\ \ldots\ ;\ W_{n-1}},$$

$$W_i = \overbrace{V_0\ ;\ V_1\ ;\ \ldots\ ;\ V_{m-1}},\ i \in \overbrace{0\ ;\ 1\ ;\ \ldots\ ;\ n-1}\ ,$$

$$V_i \in M = \overbrace{S\ ,\ E\ ,\ P\ ,\ C^s\ ,\ C^e\ ,\ C^p\ ,\ U\ ,\ *}\ ,$$

$$j \in \overbrace{0\ ;\ 1\ ;\ \ldots\ ;\ m-1},$$

де *S, E, P; C^s, C^e, C^p, U* – граматика опису операцій секвентування, елімінування, паралелення, циклічного секвентування, циклічного елімінування, циклічного паралелення і унітерма, відповідно, *a** – порожній унітерм.

Синтаксис і семантика опису унітермів

Унітерми є одного із трьох типів:

$$U \in \overbrace{U_1,\ U_2,\ U_3},$$

де U_1 – тривіальний (несеквенційний) унітерм, U_2 – секвенційний унітерм, U_3 – подвійний унітерм.

1. Опис тривіальних (несеквенційних) унітермів має такий вигляд:

$$U_1 = \overbrace{<;\ uniterm\ ;\ \#\ ;\ u\ ;\ \#\ ;\ />},$$

де *uniterm* – ідентифікатор опису унітерма, *u* – унітерм (будь-які символи та їх послідовності).

2. Такою формулою описуються секвенційні унітерми:

$$U_2 = <;\ \textit{uniterms}\ ;\ \#;\ u\_1;\ \#;\ u\_2\ ;\ \#\ ;\ />\ ,$$

де *uniterms* – ідентифікатор опису секвенційного унітерма, *u_1* – знак (позначення секвенційного унітерма) і *u_2* – секвенційні змінні унітерма.

3. Формат подвійних унітермів:

$$U_3 = <;\ \underline{unitermp};\ \#;\ u\_1;\ \#\ ;\ u\_2;\ \#;\ u\_3\ \ ;\ \#;\ />\ ,$$

де *unitermp* – ідентифікатор подвійного унітерма, *u_1* – знак подвійного унітерма, *u_2* – знак секвенційного унітерма і *u_3* – секвенційні змінні унітерма.

Опис складових операцій

А) *Секвентування* починається і закінчується такими секвенціями:

$$S_1 = <\ ;\ s_0\ ;\ \#\ ;\ r\ ;\#;\ o\ ;\ >\ \ ,$$
$$S_2 = </;\ s_0\ ;\ >\ \ ,$$

де s_0 = *"sequence"* – ідентифікатор опису секвентування, *r* – розділювач унітермів, *o* – орієнтація знаку операції секвентування,

$$z \in R = \text{“semicolon”, “coma”}\ ,$$
$$o \in O = \text{“horizontal”, “vertical”}\ ,$$

де *"semicolon"* – розділювач крапка з комою, *"coma"* – розділювач кома, *"horizontal"* – орієнтація горизонтальна, *"vertical"* – орієнтація вертикальна.

Б) *Елімінування* має такий початок і кінець опису:

$$E_1 = <\ ;\ e_0\ ;\ \#\ ;\ o\ ;\ >\ \ ,$$
$$E_2 = </;\ e_0\ ;\ >\ \ ,$$

де e_0 = *"elimination"* – ідентифікатор опису елімінування.

В) *Паралелення* починається і закінчується такими секвенціями:

$$P_1 = <\ ;\ p_0\ ;\ \#\ ;\ r\ ;\#;\ o\ ;\ >\ \ ,$$
$$P_2 = </;\ p_0\ ;\ >\ \ ,$$

де p_0 = *"parallelization"* – ідентифікатор опису паралелення.

Г) *Циклічне секвентування* має такі секвенції:

$$C^s_1 = <\ ;\ c^s\ ;\ \#\ ;\ o\ ;\ >\ \ ,$$
$$C^s_2 = </;\ c^s\ ;\ >\ \ ,$$

де c^s = *"cyclic-sequence"* – ідентифікатор опису циклічного секвентування.

Ґ) *Циклічне елімінування* починається і закінчується такими секвенціями:

$$C^e_1 = <\ ;\ c^e\ ;\ \#\ ;\ o\ ;\ >\ \ ,$$
$$C^e_2 = </;\ c^e\ ;\ >\ \ ,$$

де c^e = *"cyclic-elimination"* – ідентифікатор опису циклічного елімінування.

Д) *У циклічне паралелення* входять такі секвенції:

$$C^p_1 = <\ ;\ c^p\ ;\ \#\ ;\ o\ ;\ >\ \ ,$$
$$C^p_2 = </;\ c^p\ ;\ >\ \ ,$$

де c^p = *"cyclic-parallelization"* – ідентифікатор опису циклічного паралелення.

Синтаксис і семантика операцій

А) *Секвентування:*

$$S = S_1\ ;\ W_i;W_j\ ;\ S_2\ \ .$$

Б) *Елімінування:*

$$E = E_1\ ;\ W_i;W_j;W_k\ ;\ E_2 .$$

В) *Паралелення:*

$$P = P_1\ ;\ W_i;W_j;\ P_2 .$$

Г) *Циклічного секвентування:*

$$C^s = C^s_1\ ;\ \#\ ;\ W_g\ ;\ \#\ ;\ W_i;\ \#\ ;\ C^s_2\ \ .$$

Ґ) *Циклічного елімінування:*

$$C^{e} = \overbrace{C^{e}_{1}\,;\,\#\,;\,W_{g}\,;\,\#\,;\,W_{i};\,\#\,;\,C^{e}_{2}}\ .$$

Д) *Циклічного паралелення*:

$$C^{p} = \overbrace{C^{p}_{1}\,;\,\#\,;\,W_{g}\,;\,\#\,;\,W_{i};\,\#\,;\,C^{p}_{2}}\ ,$$

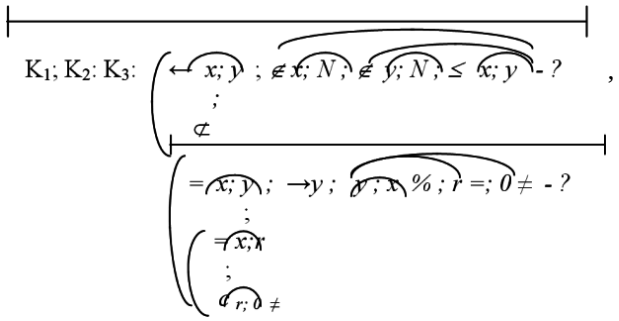
де

$$W_{g} \in u, \quad W_{j} = \overbrace{V_{0}\,;\,V_{1}\,;\,\ldots\,;\,V_{k-1}},$$

$$j \in \overbrace{0\,;\,1\,;\,\ldots\,;\,k-1}\ .$$

ПРИКЛАД ОПИСУ АЛГОРИТМУ ЕВКЛІДА

З використанням розширеної нотації формула алгоритму Евкліда [2] для обчислення двох натуральних чисел матиме такий вигляд:



де K_1 – повідомлення про те, що змінна x не є натуральним числом, N – множина натуральних чисел; K_2 – повідомлення «Значення y не є натуральним числом», K_3 – повідомлення $x \le y$, $\leftarrow x$; y – унітерм введення з клавіатури значень змінних x і y; $\notin x$; N – перевірка неналежності значення змінної x до натуральних чисел; $\notin y$; N – перевірка належності значення змінної y до натуральних чисел, $= x$; y, приписування змінній x значення змінної y, y; x % – знаходження остачі від ділення x на y;

$\overbrace{y\,;\,x\,\%\,;\,r\,=;\,0\neq\,-\,?}$ – приписування змінній остачі від ділення і її порівняння з нулем; $c_{r;\,0\neq}$ – повернення у цикл за умовою $r \neq 0$, знак операції початку циклічного секвентування.

З використанням створеної моделі граматики текстового опису ця формула опишеться так:

```
<? xml version="1.0" encoding="utf-8"?>
<root>
<elimination orientation="horizontal">
<uniterm> K1; K2: K3</uniterm>
<sequence separator="semicolon" orientation="vertical">
<uniterms></uniterms>
<sequence separator="semicolon" orientation="horizontal">
<uniterm> x </uniterm>
<uniterm> y </uniterm>
</sequence>
<cyclic-sequence orientation="vertical">
<uniterm></uniterm>
<elimination orientation="horizontal">
<sequence separator="semicolon" orientation="vertical">
<uniterms>=</uniterms>
<sequence separator="semicolon" orientation="horizontal">
<uniterm> x </uniterm>
<uniterm> y </uniterm>
</sequence>
<sequence separator="semicolon" orientation="vertical">
<uniterms>=</uniterms>
<sequence separator="semicolon" orientation="horizontal">
<uniterm> y </uniterm>
<uniterm> r </uniterm>
</sequence>
<uniterms>c</uniterms>
<sequence separator="semicolon" orientation="horizontal">
<uniterm> r </uniterm>
<uniterm> 0 </uniterm> </uniterm>
</sequence>
<uniterms></uniterms>
</sequence>
</sequence>
<uniterms></uniterms>
<uniterms>y</uniterms>
<sequence separator="semicolon" orientation="horizontal">
<sequence separator="semicolon" orientation="horizontal">
<sequence separator="semicolon" orientation="horizontal">
<uniterm> y </uniterm>
<uniterm> x </uniterm>
</sequence>
<uniterms>%</uniterms>
<uniterms>r</uniterms>
</sequence>
<uniterms>=</uniterms>
<uniterm> 0 </uniterm>
</sequence>
<uniterms></uniterms>
</elimination>
</cyclic-sequence>
</sequence>
<sequence separator="semicolon" orientation="horizontal">
<uniterms>ь</uniterms>
<sequence separator="semicolon" orientation="horizontal">
<uniterm>x</uniterm>
<uniterm>N</uniterm>
</sequence>
<sequence separator="semicolon" orientation="horizontal">
<uniterms>ь</uniterms>
<sequence separator="semicolon" orientation="horizontal">
<uniterm>y</uniterm>
<uniterm>N</uniterm>
</sequence>
<uniterms>J</uniterms>
<sequence separator="semicolon" orientation="horizontal">
<uniterm>x</uniterm>
<uniterm>y</uniterm>
```

```
</sequence>
</sequence>
</sequence>
</elimination>
</root>
```

Версія і формат кодування описані у першому зверху рядку представленого ***xml***-подібного формату. Другий рядок має назву кореневого дескриптора. Опис елімінування починається дескриптором третього рядка. У четвертому рядку описані унітерми K_1, K_2 і K_3. Опис дескриптора секвентування починається у п'ятому рядку. Шостий рядок містить секвенційний унітерм введення значень змінних з клавіатури (←). Самі ж назви змінних (*x*, *y*), яким приписуються введені з клавіатури значення, описані у сьомому, восьмому, дев'ятому і десятому рядках. Подальші рядки поданого прикладу досить легко читаються і тому не потребують додаткових пояснень.

ВИСНОВКИ

1. Створена модель розширеної нотації забезпечує текстовий опис формул алгоритмів як із тривіальними, так і секвенційними та дубльованими унітермами.

2. У моделі розширеної нотації міститься опис впорядкувань не тільки самих унітермів, а також і змінних унітермів.

3. Модель розширеної нотації забезпечує переносимість формул алгоритмів.

СПИСОК ЛІТРАТУРИ

1. *Turing, A. M.* On computable numbers, with an application to the Entscheidungsproblem / Turing A. M. // Proceedings of London Mathematical Society. – 1936–1937. – Series 2, Vol. 42. – 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].
2. *Post, E. L.* Finite Combinatory Processes - Formulation 1 / Post E. L. // Journal of Symbolic Logic. – 1936. – 1. – Pp. 103–105. – Reprinted in The Undecidable, pp. 289ff.
3. *Kolmogorov, A. N.* On the concept of algorithm (in Russian) / Kolmogorov A. N. // Uspekhi Mat. Nauk. – 1953. – 8:4. – Pp. 175–176. – [Translated into English in Uspensky V. A., Semenov A. L. Algorithms: Main Ideas and Applications. – Kluwer, 1993.]
4. *Aho, A. V.* The design and analysis of computer algorithms / Aho A. V, Hopcroft J. E, Ullman J. D. – Addison-Wesley Publishing Company, 1974.
5. *Schönhage, A.* Universelle Turing Speicherung / Schönhage A. // In J. Dörr and G. Hotz, Editors, Automatentheorie und Formale Sprachen. – Bibliogr. Institut, Mannheim, 1970. – Pp. 369–383.
6. *Church, A.* An unsolvable problem of elementary number theory / Church A. // American Journal of Mathematics. – 1936. – Vol. 58. – Pp. 345–363.
7. *Markov, A. A.* Theory of algorithms (in Russian) / Markov A. A. // Editions of Academy of Sciences of the USSR. – 1951. – Vol. 38. – Pp. 176–189 – [Translated into English in American Mathematical Society Translations, 1960, series 2, 15, pp. 1–14.]
8. *Krinitski, N. A.* Algorithms around us (in Russian) / Krinitski N. A. – Moscow : Mir, 1988. – [Also translated to Spanish (Algoritmos a nuestro alrededor)].
9. *Успенский, В. А.* Теорія алгоритмов: основные открытия и приложения / Успенский В. А., Семенов А. Л. – М. : Наука, 1987. – 288 с.
10. *Овсяк, В.* Засоби еквівалентних перетворень алгоритмів / Овсяк В. // Доповіді національної академії наук України. – 1996. – № 9. – С. 83–89.
11. *Owsiak, W.* Rozszerzenie algebry algorytmyw / Owsiak W., Owsiak A. // Pomiary, automatyka, kontrola. – № 2. – S. 184–188.
12. *Дейтел, Х.* C# / Дейтел Х. и др. – М. ; СПб : БХВ-Петербург, 2006. – 1056 с.
13. *Мак-Дональд, М.* WPF. Windows Presentation Foundation в NET 3.5 с примерами на 2008 / Мак-Дональд М. – Второе издание. – М. : Вильямс, 2008. – 928 с.
14. Великий тлумачний словник сучасної української мови / уклад. і голов. ред. В. Т. Бусел. – К. ; Ірпінь: Перун, 2002. – 1440 с.
15. *Бритковський, В. М.* Моделювання редактора формул секвенцій них 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.
16. *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.
17. *Овсяк, О. В.* Класи інформаційної системи генерування коду / Овсяк О. В. // Вісник Тернопільського державного технічного університету. – 2010. – № 1. – С. 171–176.

Надійшла 06.07.2010

Овсяк А. В.

МОДЕЛЬ РАСШИРЕННОЙ НОТАЦИИ ТЕКСТОВОГО ОПИСАНИЯ ФОРМУЛ АЛГОРИТМОВ

С целью записи в память компьютера графически-текстовых формул алгоритмов, которые образованы функциональными унитермами с упорядоченными переменными и параметрами, создано расширенную **xml**-подобную нотацию описания формул алгоритмов. Модель расширенной нотации описана средствами алгебры алгоритмов с использованием операции секвенцирования.

Приведен пример использования расширенной нотации для описания формулы алгоритма Евклида.

Ключевые слова: модель, нотация, синтаксис, семантика, унитерм, секвенцирование, элиминирование, параллирование.

Ovsyak O. V.

MODEL OF EXTENDED NOTATION OF ALGORITHM FORMULAS TEXTUAL DESCRIPTION

The proposed extended **xml**-notation of algorithm formulas description provides writing of graphical-textual algorithm formulas formed by functional uniterms with ordered variables and parameters in the computer memory. The extended notation model is described by means of algorithm algebra using the sequencing operation.

The example of Euclid algorithm formula description using the extended notation is given in the paper.

Key words: model, notation, syntax, semantics, uniterm, sequencing, elimination, paralleling.