ISSN 2079-3316 ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ №3(26), 2015, с. 45–52

УДК 519.681.2

А. А. Демидов

О пополнении группоида до программной алгебры

Аннотация. Задача вложения конечного группоида в конечную программную алгебру имеет прикладное значение для преобразования алгоритма в форму, пригодную для вычисления на алгебраическом процессоре. Она была поставлена и решена Н. Н. Непейводой для полугрупп, затем им же было построено вложение группоида в бесконечную программную алгебру. В данной работе строится вложение конечного группоида в конечную программную алгебру, что завершает решение указанной задачи.

Ключевые слова и фразы: алгебры, алгебраические вычисления, вложение группоида.

Программные алгебры GAPS (Generic algebraic program structure) были предложены Н. Н. Непейводой [1] в рамках развития алгебраических методов представления программных систем.

Задача вложения конечного группоида в конечную программную алгебру имеет прикладное значение для преобразования алгоритма в форму, пригодную для вычисления на алгебраическом процессоре. Она была поставлена и решена в работе [1] для полугрупп, затем было построено вложение группоида в бесконечную программную алгебру [2].

В данной работе строится вложение конечного группоида в конечную программную алгебру, что завершает решение задачи.

Программная алгебра — бигруппоид с ассоциативной операцией «$\circ$» композиции программ и неассоциативной операцией «$*$» аппликации (применения программы к данным), которые связаны тождествами:

$$(x * f) * g = x * (f \circ g);\quad 0 * x = x * 0 = 0;\quad x * e = x. \tag{1}$$

Проект проводится при финансовой поддержке государства в лице Минобрнауки России (идентификатор № RFMEFI61314X0030).

Искомое вложение будем строить с привлечением конечных автоматов, получающих на вход запись выражений над элементами конечного группоида.

С «переходной системой» (A, Q, φ) конечного автомата над входным алфавитом A с множеством состояний Q и функцией переходов $\varphi\colon Q \times A \to Q$ можно связать полугруппу подстановок на множестве Q [3, 4]: каждой букве $a \in A$ на основе функции переходов поставим в соответствие подстановку

$$\varphi_a\colon Q \to Q, \quad \varphi_a(q) = \varphi(q, a)$$

перехода автомата из предыдущего состояния q в следующее. Последовательному действию букв a и b соответствует композиция

$$\varphi_{ab} = \varphi_b(\varphi_a(q)) = \varphi_a \circ \varphi_b. \tag{2}$$

Множество подстановок $\{\varphi_\alpha : \alpha \in A^*\}$ всех слов A^* в алфавите A образует полугруппу $P_{\mathcal{A}}$, называемую внутренней полугруппой автомата $\mathcal{A}$. Она изоморфна фактор-полугруппе полугруппы A^* с операцией конкатенации (соединения слов) по отношению

$$R = \{(\alpha, \beta)\colon \alpha, \beta \in A^* \quad \text{и} \quad \forall q \in Q\ (\varphi(q, \alpha) = \varphi(q, \beta))\}.$$

Указанный изоморфизм позволяет связать формальную запись выражений над элементами конечного группоида G с операциями над элементами внутренней полугруппы соответствующего автомата, далее эту связь несложно распространить и на конечные программные алгебры.

Определение 1. Группоид G назовём редуктивным слева, если из того, что $x \cdot a = x \cdot b$ для всех $x \in G$, следует $a = b$ $(a, b \in G)$ — см. [5].

Лемма 1 (Основная). *Каждый редуктивный слева конечный группоид допускает вложение во внутреннюю полугруппу конечного автомата.*

Формула (2) устанавливает соответствие между выражениями вида $a \cdot b$ над элементами группоида G и композициями вида $\varphi_a \circ \varphi_\bullet \circ \varphi_b$ над элементами полугруппы $P_{\mathcal{A}}$. Для построения вложения $G \mapsto P_{\mathcal{A}}$ необходимо избавиться от подстановки $\varphi_\bullet$, соответствующей символу операции.

Гомоморфизм, задаваемый левым сдвигом внутренней полугруппы $P_{\mathcal{A}}$ конечного автомата:

$$\varphi_x \mapsto \varphi_{\bullet x} = \varphi_\bullet \circ \varphi_x \tag{3}$$

— позволяет опустить знак операции в записи алгебраических выражений, что обычно и делается. Формально можно записать:

$$ab \quad \Leftrightarrow \quad \varphi_\bullet^{-1} \circ \varphi_\bullet \circ \varphi_a \circ \varphi_\bullet \circ \varphi_b \;=\; \varphi_\bullet^{-1} \circ \varphi_{\bullet a} \circ \varphi_{\bullet b}, \tag{4}$$

где $\varphi_\bullet^{-1}$ — формальная запись, определяющая тождество $\left[\varphi_\bullet^{-1} \circ \varphi_\bullet\right] \circ \varphi_a \equiv \varphi_a$; в общем случае элемента $\varphi_\bullet^{-1}$ не существует, поскольку $\varphi_\bullet$ не обязательно является биекцией. Это не мешает нам с помощью некоторого символа $\#$ перевести конечный автомат в особое «начальное» состояние, когда последующий аргумент x на входе будет действовать как подстановка φ_x, а не $\varphi_{\bullet x}$. Формально это означает вложение исходного группоида G в группоид $G_\#$ с удвоенным множеством элементов.

Определение 2. Назовём элементы $\#x \Leftrightarrow \varphi_x$ удвоенного группоида оригинальными, а элементы $x \Leftrightarrow \varphi_{\bullet x}$ — сопряжёнными к ним.

Подстановка $\varphi_\#$, соответствующая входному символу $\#$, по построению — идемпотент: дважды перейдя в начальное состояние, окажемся там же. Руководствуясь выражением (4), формально можно записать:

$$\varphi_\#^2 = \varphi_\# = \varphi_\bullet^{-1} \circ \varphi_\bullet .$$

Конечный автомат тогда можно представить как функцию $F : 2^{(G_\#)} \to P_{\mathcal{A}}$ из множества наборов аргументов в множество подстановок, действующую следующим образом:

$$\begin{gathered} F(\#a) = \varphi_\# \circ \varphi_a, \quad F(a) = \varphi_{\bullet a}, \quad F(b) = \varphi_{\bullet b}, \\ F(\#ab) = \varphi_\# \circ \varphi_a \circ \varphi_{\bullet b}. \end{gathered} \tag{5}$$

Пусть результат произведения ab в выражении (5) равен c. Формально выполним следующие преобразования:

$$\begin{gathered} F(\#ab) = \varphi_\# \circ \varphi_a \circ \varphi_{\bullet b} = \varphi_c; \\ \varphi_\bullet \circ F(\#ab) = \varphi_\bullet \circ \varphi_\bullet^{-1} \circ \varphi_\bullet \circ \varphi_a \circ \varphi_{\bullet b} = \varphi_\bullet \circ \varphi_c; \\ \varphi_{\bullet a} \circ \varphi_{\bullet b} = \varphi_{\bullet c} \end{gathered} \tag{6}$$

— откуда следует, что подстановка $\varphi_\bullet^{-1}$ в выражении (4) является левым сдвигом $\varphi_{\bullet x} \mapsto \varphi_x$, возвращающим результат от сопряжённого к оригинальному виду.

Предложение 1. *Если $\varphi_\bullet$ — биекция, то F — изоморфизм, определённый на сопряжённых элементах удвоенного группоида.*

Доказательство. Так как язык A^* распознаётся автоматом, и все оригинальные элементы удвоенного группоида различны, то на этих элементах функция F является однозначной. В силу биективности $\varphi_\bullet$ левый сдвиг (3) является изоморфизмом, поэтому F будет однозначной и на сопряжённых элементах. Отсюда, с учётом выражений (5) и (6), следует, что F — изоморфизм, определённый на сопряжённых элементах:

$$F(a) \circ F(b) = F(ab).$$

Оригинальным элементам сопоставлены подстановки φ_x, и они указанному изоморфизму, вообще говоря, не удовлетворяют. □

Предложение 2. *Функция $\varphi_\bullet$ является биекцией тогда, когда G — редуктивный слева конечный группоид.*

Доказательство. Пусть $\varphi_\bullet$ не является биекцией, то есть отображает некоторые элементы $a \neq b$ в одну подстановку $\varphi_{\bullet a} \equiv \varphi_{\bullet b}$. Тогда, согласно выражениям (5), для любого элемента x будет верно:

$$F(\#xa) = \varphi_\# \circ \varphi_x \circ \varphi_{\bullet a} \equiv \varphi_\# \circ \varphi_x \circ \varphi_{\bullet b} = F(\#xb).$$

В силу регулярности языка A^* если двум выражениям сопоставлена одна и та же подстановка, то эти выражения равны. Поэтому равенство $F(\#xa) = F(\#xb)$ влечёт равенство $x \cdot a = x \cdot b$ для всех x. □

Замечание 1. *Обратное неверно: подстановки $\varphi_{\bullet a} = (2\ 2\ 1\ 2)$, $\varphi_{\bullet b} = (2\ 2\ 4\ 2)$ и $\varphi_{\bullet c} = (2\ 2\ 2\ 2)$ дают контрпример $x \cdot a = x \cdot b = c$.*

Доказательство основной леммы. Доказательство сводится к доказательству предложений 1 и 2. □

Формулировка леммы 1 ограничивает возможности метода. Пример 1 демонстрирует вложение конечного группоида в конечную программную алгебру, которое не может быть построено предложенным методом.

Пример 1. К полугруппе S с нулевым умножением $x \cdot y = 0$ для всех $x, y \in S$ присоединим правую единицу: $x \cdot e = x$, $e \cdot x = 0$. Полученный группоид G не является редуктивным слева. Определим на G программную алгебру с операциями аппликации $x * y = x \cdot y$ и композиции

$$x \circ y = \begin{cases} x, & \text{при } x \neq e; \\ y, & \text{иначе.} \end{cases}$$

Группоид G вкладывается в эту алгебру по операции аппликации.

Указанный недостаток общности метода является устранимым — достаточно предварительно расширить исходный группоид новыми элементами так, чтобы выполнялось условие предложения 2.

Предложение 3. *Каждый конечный группоид допускает вложение в редуктивный слева конечный группоид.*

Доказательство. Пусть группоид G содержит элементы $a \neq b$, такие что $x \cdot a = x \cdot b$ для всех $x \in G$. Расширим его до группоида G', определив новый элемент $e \in G'$ как единицу в G', так что $e \cdot x = x \cdot e = x$ для всех $x \in G'$. Тогда $e \cdot a = a \neq b = e \cdot b$ для любых $a, b \in G'$, таких что $a \neq b$. □

Теорема 1. *Каждый конечный группоид допускает вложение в конечную программную алгебру.*

Доказательство. В соответствии с леммой 1 и предложением 3 каждый конечный группоид вкладывается в конечную полугруппу. На элементах полугруппы $P_{\mathcal{A}}$ определим вторую операцию «*» так, чтобы соблюдались аксиомы программной алгебры (1).

Так, можно взять фиксированный идемпотент $d \in P_{\mathcal{A}}$ и положить $x * y = d \circ x \circ y$, тогда первая аксиома будет выполнена, поскольку

$$(x * y) * z = d \circ (d \circ x \circ y) \circ z = d^2 \circ x \circ (y \circ z) = x * (y \circ z);$$

вторая аксиома выполняется при наличия нуля в $P_{\mathcal{A}}$, поскольку

$$\begin{matrix} (x * 0) * y = x * (0 \circ y) = x * 0 \\ (x * y) * 0 = x * (y \circ 0) = x * 0 \end{matrix} \quad \Rightarrow \quad 0 \circ y = y \circ 0 = 0;$$

третья аксиома выполняется при наличия правой единицы в $P_{\mathcal{A}}$, поскольку

$$(x * y) * e = x * (y \circ e) = x * y \quad \Rightarrow \quad y \circ e = y.$$

Нуль и правую единицу в случае их отсутствия можно присоединить к полугруппе $P_{\mathcal{A}}$, как это было проделано в доказательстве предложения 3 в отношении группоида G. $\square$

В заключение необходимо отметить, что вложение в соответствии с предложенным алгоритмом не является минимальным. Так, пример 1 демонстрирует вложение, не требующее добавления новых элементов.

Благодарности

Автор выражает признательность Н. Н. Непейводе за оказанное доверие, формулировку задачи и ценные советы, способствовавшие решению.

Список литературы

[1] Н. Н. Непейвода, «От численного моделирования к алгебраическому», *Труды VI Международной конференции «Параллельные вычисления и задачи управления»*. Т. 1, PACO-2012 (Москва, 24–26 октября 2012 г.), Ин-т проблем управления им. В. А. Трапезникова РАН, с. 93–103 ↑ 45.

[2] Н. Н. Непейвода. «Алгебраический подход к управлению», *Проблемы управления*, 2013, №6, с. 2–14 ↑ 45.

[3] С. В. Алёшин. «Полугруппы и группы автоматов», *Интеллектуальные системы*, **17**:1–4 (2013), с. 129–141 ↑ 46.

[4] В. М. Глушков. «Абстрактная теория автоматов», *Успехи мат. наук*, **16**:5(101) (1961), с. 3–62 ↑ 46.

[5] А. Клиффорд, Г. Престон. *Алгебраическая теория полугрупп*. Т. 1, Мир, М., 1972, 283 с. ↑ 46.

Рекомендовал к публикации *д.ф.-м.н. Непейвода Н. Н.*

Об авторе:

Алексей Александрович Демидов
Младший научный сотрудник Исследовательского центра искусственного интеллекта ИПС им. А. К. Айламазяна РАН.
e-mail: alex@dem.botik.ru

Пример ссылки на эту публикацию:

А. А. Демидов. «О пополнении группоида до программной алгебры», *Программные системы: теория и приложения*, 2015, **6**:3(26), с. 45–52.
URL http://psta.psiras.ru/read/psta2015_3_45-52.pdf

Aleksey Demidov. *To the completion of a groupoid to a programm algebra.*

Abstract. The task of embedding of a finite groupoid into a finite programm algebra has a practical significance for the conversion of algorithm into a form suitable for computation on algebraic processor. It was posed and solved by N. N. Nepeivoda for semigroups, then he has proposed a method of embedding of a groupoid into a infinite programm algebra. In this paper we construct an embedding of a finite groupoid into a finite programm algebra, which completes the solution of the problem. *(In Russian).*

Key Words and Phrases: algebras, algebraic computations, groupoid embedding.

References

[1] N. N. Nepeyvoda, “Ot chislennogo modelirovaniya k algebraicheskomu”, *Trudy VI Mezhdunarodnoy konferentsii “Parallel’nyye vychisleniya i zadachi upravleniya”*. V. 1, PACO-2012 (Moskva, 24–26 oktyabrya 2012 g.), In-t problem upravleniya im. V. A. Trapeznikova RAN, pp. 93–103 (in Russian).

[2] N. N. Nepeyvoda. “Algebraicheskiy podkhod k upravleniyu”, *Problemy upravleniya*, 2013, no.6, pp. 2–14 (in Russian).

[3] S. V. Aleshin. “Polugruppy i gruppy avtomatov”, *Intellektual’nyye sistemy*, **17**:1–4 (2013), pp. 129–141 (in Russian).

[4] V. M. Glushkov. “The abstract theory of automata”, *Russian Math. Surveys*, **16**:5 (1961), pp. 1–53.

[5] A. H. Clifford, G. B. Preston, *The algebraic theory of semigroups*. V. 1, Mathematical Surveys, vol. **7**, AMS, Providence, RI, 1961.

Sample citation of this publication:

Aleksey Demidov. “To the completion of a groupoid to a programm algebra”, *Program systems: theory and applications*, 2015, **6**:3(26), pp. 45–52. (*In Russian.*)

URL http://psta.psiras.ru/read/psta2015_3_45-52.pdf