

О ПОСТРОЕНИИ КЛООНОВ АЛГЕБР ФУНКЦИОНАЛЬНЫХ n -ОТНОШЕНИЙ

Л.М. Захария, Т.В. Луценко, Г.Е. Цейтлин

Институт программных систем НАН Украины, 03187, Киев, пр. Академика Глушкова, 40
tseytlin@vikno.relc.com

Рассматриваются клоны алгебр n -отношений. Данные клоны включают алгебры, равномошные алгебре Кодда. Особое внимание уделено распространению результатов, полученных ранее по полугрупповым (грамматическим и алгоритмическим) клонам на построенные клоны алгебр функциональных n -отношений; эти клоны определяются посредством соответствующих суперпозиций n -отношений и связанных с ними функций, соответственно. Очерчены одномерные и многомерные вычислительные структуры и ассоциированные с ними предметные области.

Clones of algebras of n -relations are examined. The given clones include algebras, equipotent to Codd algebra. High emphasis is placed on distribution of the results received on semigroup (grammatical and algorithmic) clones to the clones of algebras of functional n -relations. Both one-dimensional and multivariate computing structures and the subject domains associated with them are outlined.

Введение

Понятие клона относится к числу фундаментальных понятий универсальной и общей алгебры [1, 2]. Наряду с традиционным его использованием при функциональных построениях в многозначных логиках [3-5] понятие клона приобретает теоретический и прикладной интерес для алгоритмических и полугрупповых построений в алгоритмике [6, 7].

Под клоном понимается, обычно, одноосновная универсальная алгебра вида $K = \langle A; SUPER \rangle$, где A – основа, представляющая собой множество однотипных функций; $SUPER$ – сигнатура, состоящая из операции суперпозиции функций (подстановки одних функций вместо переменных в другие), а также операций отождествления и переименования переменных функции (более детально различные определения суперпозиции см. [3-5, 8, 9]).

Данная статья посвящена построению клонов алгебр функциональных n -отношений [8, 10, 11]. Следует отметить важное значение алгебр n -отношений [10] в реляционных базах данных и знаний [12, 13]. Функциональные n -отношения весьма важны и при проектировании различных объектов, характерных для построений в методологии и технологии современного программирования. Это, в частности, связано с ответами на вопросы "что?" и "как?" Ответ на первый вопрос сопряжен с точной постановкой решаемой задачи и может быть формализован в терминах теории отношений; а второй – с построением функций для её решения [11]. Дальнейшее развитие теории отношений имеет важное значение и для других перспективных разделов прикладного программирования [12, 13].

Материал статьи подчинён следующей структуре. В разд. 1 приведен ряд распространённых операций над n -отношениями: *композиция* бинарных отношений; *свёртка де Моргана*; *транспозиция*, *отождествление* и *переименование столбцов* и др.

Разд. 2 посвящён операции i -суперпозиции [8], которая является естественным обобщением на n -арный случай композиции бинарных отношений. В частности, отмечается связь между i -суперпозицией и функциональностью. Функциональные построения однородных и неоднородных преобразований вычислительных структур различной размерности приведены в разд. 3. Наконец, в заключении подведены итоги выполненного исследования и приведены постановки некоторых задач, определяющих перспективы дальнейшего развития работ по алгебре алгоритмики.

1. Операции над n -отношениями

Одной из распространённых операций теории бинарных отношений следует считать их композицию, состоящую в последовательном выполнении бинарных отношений.

Пусть R и G – бинарные отношения, определённые на множестве M , $R, G \subseteq M^2$, где M^2 – декартова степень M . Под композицией $R * G$ понимается новое бинарное отношение $W = R * G$, состоящее из пар вида $(a, s) \in W$ тогда и только тогда, когда в M найдётся хотя бы один промежуточный элемент b такой, что пара $(a, b) \in R$, а пара $(b, s) \in G$. Таким образом, W суть последовательное вычисление R и G .

Бинарное отношение R – функционально, если с ним связана функция $f(x)$ так, что для любой пары $(a, b) \in R$ выполняется $f(a) = b$. При этом если R и G – функциональны и с ними связаны функции

$f(x)$ и $g(y)$, то с композицией $W = R * G$ связана функция $w(u)$, которая служит суперпозицией функций f и g , $w(x) = g(f(x))$. Иными словами, вычисление значения функции $w(a) = s$ состоит в вычислении функции $f(a) = b$, а затем функции $g(b) = s$, т.е. значение функции w состоит в последовательном вычислении функции f , а затем g .

Аналогом композиции для n -арных отношений ($n \geq 2$) служит свёртка де Моргана [10]. Под n -арным отношением (кратко, n -отношением) R^n понимается подмножество n -мерного декартова произведения, $R^n \subseteq M_1 * M_2 * \dots * M_n$. В частности, указанные множества могут быть равны $M_1 = M_2 = \dots = M_n$, тогда подразумевается декартова степень M . Примерами n -отношения могут служить ведомости, часто используемые в экономических задачах. Например, информация о сотруднике может быть описана следующим n -отношением: $V^c \subset M_1 * M_2 * M_3 * M_4$,

где M_1 – множество табельных номеров сотрудников;

M_2 – множество фамилий сотрудников;

M_3 – множество дневных тарифов сотрудников;

M_4 – множество, состоящее из возможных размеров социальных льгот по налогу на доходы физических лиц,

а табель рабочих дней может быть представлен таким n -отношением: $T^c \subset M_1 * M_5$, где M_5 – множество целых чисел из интервала $[0, 23]$, фиксирующее количество отработанных сотрудником дней в месяце.

Пусть $R^k \subseteq M_1 * M_2 * \dots * M_k$ и $G^c \subseteq L_1 * L_2 * \dots * L_c$ суть n -арные отношения, определённые на декартовых произведениях указанных множеств, где $M_k \cap L_1 \neq \emptyset$ – непустое пересечение множеств M_k и L_1 . Свёртка де Моргана определяется следующим образом: результирующее $(k + c - 1)$ -арное отношение $W^{(k+c)-1} = R^k * G^c$ представляет собой совокупность кортежей вида

$$W^{(k+c)-1} = \{ \tilde{z} \mid \tilde{z} : (a_1, a_2, \dots, a_{n-1}, b, q_2, \dots, q_c) \text{ и } b \in M_k \cap L_1 \},$$

здесь b – промежуточный элемент такой, что $(a_1, a_2, \dots, a_{n-1}, b) \in R^k$, $(b, q_2, \dots, q_n) \in G^c$. Данная операция формализует процесс связывания двух таблиц баз данных по значению поля b . Для нашего примера свертка де Моргана определит новое отношение $W^5 = V^c * T^c$, которое опишет структуру исходной информации при решении задачи начисления заработной платы в виде следующей ведомости:

Табельный номер	Фамилия сотрудника	Дневной тариф	Количество социальных льгот	Количество отработанных дней
-----------------	--------------------	---------------	-----------------------------	------------------------------

Положим, что n -отношение R^n функционально по n -й компоненте, если с ним связана $n-1$ -местная функция $f(x_1, x_2, \dots, x_{n-1})$, областью значений которой служит множество M_n . Так, для любого v_{n-1} кортежа значений переменных данной функции она принимает значение в множестве M_n , $f(v_{n-1}) \in M_n$. Иллюстрацией функционального отношения служит отношение $R^6 \subset M_1 * M_2 * M_3 * M_5 * M_6$, где M_6 – множество начисленной заработной платы каждого сотрудника, определяемое функцией $f_{\text{начисление}}(w) = \langle \text{дневной тариф} \rangle * \langle \text{количество отработанных дней} \rangle = \langle \text{начислено} \rangle$, $w \in W^5$.

Пусть n -отношения R^k и G^c – функциональны и с ними связаны функции $f(x_1, x_2, \dots, x_{k-1})$ и функция $g(x_1, x_2, \dots, x_{c-1})$; тогда в результате свёртки де Моргана получим также функциональное $((k + c) - 1)$ -арное отношение $W^{(k+c)-1} = R^k * G^c$, с которым связана следующая суперпозиция функций:

$$h(x_1, x_2, \dots, x_{(k+c)-1}) = g(f(x_1, x_2, \dots, x_{k-1}), x_{k+1}, \dots, x_{(k+c)-1}). \quad (1)$$

Проиллюстрируем сказанное на примере расчета налога на доходы физических лиц для каждого сотрудника. Функция $g(r)$, где $r \in M_1 * M_2 * M_3 * M_5 * M_6$, определяет элементы множества M_7 значений налога на доходы физических лиц (подходящий налог составляет 13%, размер социальной льготы, не облагаемой налогом на доходы в 2005 году составил 131 грн, количество социальных льгот для каждого сотрудника опре-

деляется его социальным статусом – количество детей, инвалидность и т.п.),
 $g(r) = \langle \text{начислено} \rangle - \langle \text{количество социальных льгот} \rangle * 131 * 13\% = \langle \text{налог на доходы} \rangle$.

В результате свертки де Моргана отношений R^6 и P^2 ($P^2 \subset M^1 * M^7$) по полю <табельный номер> получим следующее n -отношение, которое для наглядности представим в виде таблицы базы данных

Табельный номер	Фамилия сотрудника	Дневной та-риф	Количество социальных льгот	Количество отработанных дней	Начислено	Налог на доходы
-----------------	--------------------	----------------	-----------------------------	------------------------------	-----------	-----------------

которое определяется следующей суперпозицией функций:

$$h(x_1, x_2, \dots, x_6) = g_{\text{налог}}(f_{\text{начислено}}(w), x_1, x_2, \dots, x_5)$$

(в выражении произведены отождествление и перестановка столбцов, $x_1 \in M^1$, $x_2 \in M^2$, $x_3 \in M^3$, $x_4 \in M^4$, $x_5 \in M^5$, $x_6 \in M^6$). Данный пример иллюстрирует процесс решения задачи начисления заработной платы в виде суперпозиции функций.

Таким образом, из проведенных построений вытекает справедливость следующего утверждения.

Теорема 1. Свёртка де Моргана $W^{(k+c)-1} = R^k * G^c$ функциональных n -отношений R^k и G^c порождает новое функциональное отношение, связанное с суперпозицией h исходных функций f и g , которая задана равенством (1).

Для n -отношений естественным образом вводятся операции перестановки, отождествления и переименования столбцов (по аналогии с введенными Мальцевым [5] одноименными операциями, которые применимы в функциональных построениях [3, 4, 8]). Тем самым справедливо утверждение.

Следствие 1. Циклическая перестановка столбцов в сочетании со свёрткой де Моргана функциональных n -отношений снова порождает функциональные n -отношения, согласно суперпозициям соответствующих этим n -отношениям функций.

В [10] рассмотрены и другие операции над n -отношениями: проекция, фильтрация, теоретико-множественные булевы операции и др., которые широко используются в алгебрах структур данных, например алгебре Кодда [12].

Следствие 2. Теория n -отношений служит математическим фундаментом в прикладном программировании при построении реляционных баз данных и знаний [12,13].

Одной из основных проблем алгебры клонов является проблема функциональной полноты. Для алгебры логики $L/2$ доказана теорема о функциональной полноте Поста и построена решетка максимальных подалгебр. Между алгеброй логики $L/2$ и алгеброй отношений $O/2$, определенной на множестве пар, состоящих из нулей и единиц, существует связь, которая позволяет сформулировать теорему о функциональной полноте для алгебры $O/2$. Для алгебры логики $L/2$ естественно рассмотреть обобщенную свёртку де Моргана, как распространение обычной суперпозиции функций на случай n -отношений, тогда из принадлежности функций, участвующих в суперпозиции, к некоторому замкнутому классу следует, что и полученная в результате суперпозиции функция также принадлежит тому же классу. Например, пусть все функции из суперпозиции сохраняют константу 0, тогда соответствующие n -отношения содержат наборы, состоящие из нулей; этот факт имеет место и для отношения, соответствующего функции h , полученной в результате суперпозиции.

Другой пример: пусть все функции f_i из суперпозиции самодвойственны, тогда и результирующая функция h также самодвойственна, но отношения, соответствующие функциям f_i , содержат противоположные наборы; этому свойству удовлетворяет и отношение, соответствующее функции h .

Пусть алгебре логики $L/2$ соответствует алгебра отношений $O/2$, сигнатура операций которой содержит обобщенную свёртку де Моргана и унарные операции для отождествления и переименования переменных. Тогда для алгебры отношений $O/2$ может быть сформулирована следующая теорема.

Теорема Поста для алгебры $O/2$.

Произвольно выбранная система отношений функционально полна в алгебре $O/2$ тогда и только тогда, когда эта система содержит:

- хотя бы одно отношение, не принадлежащее подалгебре $O/0$, сохраняющих константу 0;
- хотя бы одно отношение, не принадлежащее подалгебре $O/1$, сохраняющих константу 1;
- хотя бы одно отношение, не принадлежащее подалгебре O/c самодвойственных функций;
- хотя бы одно отношение, не принадлежащее подалгебре O/m монотонных функций;
- хотя бы одно отношение, не принадлежащее подалгебре O/l линейных функций.

Следствие 1. Каждой подалгебре алгебры логики $L/2$ соответствует подалгебра алгебры отношений $O/2$.

Следствие 2. Перечисленные в сформулированной теореме Поста для алгебры $O/2$ классы отношений являются для алгебры $O/2$ максимальными и соответствуют максимальным подалгебрам алгебры $L/2$.

Следствие 3. Алгебра $O/2$ имеет счётное множество подалгебр, решётка которых соответствует диаграмме Поста для алгебры $L/2$.

$$\begin{aligned}
 & H(x_{11}, x_{12}, \dots, x_{1n_1}; x_{21}, x_{22}, \dots, x_{2n_2}; \dots; x_{m1}, x_{m2}, \dots, x_{mn_m}) = \\
 & = f(g_1(x_{11}, x_{12}, \dots, x_{1n_1}), g_2(x_{21}, x_{22}, \dots, x_{2n_2}), \dots, g_m(x_{m1}, x_{m2}, \dots, x_{mn_m})).
 \end{aligned}
 \tag{3}$$

Отметим, что наряду со свёрткой де Моргана операции перестановки столбцов, их отождествления, переименования и пр. индуцируют соответствующие операции над переменными функций, введенные для функциональных построений в многозначных логиках [4], в частности двузначной алгебре логики, детально исследованной Постом, которая образует клон Поста (КП) [6, 7].

3.2. Суперпозиции функций в преобразованиях многомерных структур. Под многомерными однородными вычислительными структурами, ориентированными на синхронные параллельные вычисления понимаются, в частности, одномерные (линейные) абстрактные регистры, односторонне или двусторонне бесконечные, а также их двухмерные (плоские) и многомерные обобщения [8].

Пусть R – двоичный линейный абстрактный регистр, двусторонне бесконечный. Периодически определенное преобразование (далее ПО-преобразование) F регистра R задаётся булевой функцией $f(x_1, x_2, \dots, x_c)$, которая синхронно вычисляет новое состояние $B = (\dots, b_{i-1}, b_i, b_{i+1}, \dots)$ регистра R по его текущему состоянию $A = (\dots, a_{i-1}, a_i, a_{i+1}, \dots)$ следующим образом:

$$b_i = f(a_i, a_{i+1}, \dots, a_{i+c-1}),$$

где b_i – новое значение ячейки r_i регистра R в состоянии B , $a_i, a_{i+1}, \dots, a_{i+c-1}$ – текущие значения ячеек окрестности в состоянии A , от которых зависит состояние произвольно выбранной ячейки r_i регистра R . Пусть заданы ПО-преобразования F и G регистра R посредством булевых функций $f(x_1, x_2, \dots, x_c)$ и $g(x_1, x_2, \dots, x_q)$ соответственно. Композиция $H = G * F$ ПО-преобразований регистра R порождает новое ПО-преобразование H данного регистра, которое определяется следующей суперпозицией функций f и g :

$$h(x_1, x_2, \dots, x_{c+q-1}) = f(g(x_1, x_2, \dots, x_q), g(x_2, x_3, \dots, x_{q+1}), \dots, g(x_c, x_{c+1}, \dots, x_{c+q-1})). \tag{4}$$

Операция композиции ПО-преобразований удовлетворяет свойству ассоциативности, тем самым совокупность ПО-преобразований на абстрактном регистре образует полугруппу [8]. Справедливы следующие утверждения.

Лемма 1. Полугруппа ПО-преобразований на абстрактном линейном регистре R не имеет конечного базиса [15].

На множестве всех булевых функций введём сигнатуру операций, включающую бинарную суперпозицию [4], а также унарные операции над переменными, предложенные в [5]. Полученную таким образом алгебру назовём модифицированным клоном Поста КП/м.

Лемма 2. Алгебра КП/м имеет конечный базис [9].

Проведенные построения допускают распространение на случай плоских регистров и вычислительных структур большей размерности.

3.3. Неоднородные алгоритмические преобразования на вычислительных структурах, ориентированных на мультиобработку. Понятие ПО-преобразования на абстрактном регистре носит однородный характер в том смысле, что его вычисление осуществляется на всём регистре посредством одной функции. Естественным обобщением однородного ПО-преобразования служит его неоднородная трактовка, предназначенная для существенного расширения сферы применимости ПО-преобразований на вычислительных структурах, ориентированных на мультиобработку.

Неоднородное ПО-преобразование P вычислительной структуры S определяется посредством системы функций $\langle f_1(x_1, x_2, \dots, x_n), f_2(x_1, x_2, \dots, x_n), \dots, f_n(x_1, x_2, \dots, x_n) \rangle$, которые заданы на окрестности $O = \{y_1, y_2, \dots, y_n\}$, распределённой по структуре S так, что

$b_1 = f_1(a_1, a_2, \dots, a_n), b_2 = f_2(a_1, a_2, \dots, a_n), \dots, b_n = f_n(a_1, a_2, \dots, a_n)$, где a_1, a_2, \dots, a_n – текущее, b_1, b_2, \dots, b_n – новое состояния элементов окрестности S .

Проиллюстрируем сказанное на задачах символьной мультиобработки, к которым, в частности, относится сортировка массивов [16-18].

Пример 1. Рассмотрим одномерную кольцеобразную вычислительную структуру КВС, содержимым которой является в общем случае неупорядоченный числовой массив $m_0 : a_1 a_2 \dots a_n$, свёрнутый в кольцо. В зависимости от длины n кольцо содержит разделители–маркеры $\$$: при нечётном n такой маркер один, а при

n чётном их два. Тем самым длина кольца КВС всегда чётна. Предполагается также, что $\$$ больше любого числа непосредственно слева от него и меньше любого числа справа.

Кольцо КВС разделено на попарно непересекающиеся окрестности-пары: значение левого элемента каждой пары (a_i, a_{i+1}) определяется функцией $\min(a_i, a_{i+1})$, тогда как её правый элемент – функцией $\max(a_i, a_{i+1})$.

Зададим на КВС неоднородное ПО-преобразование TRANSP-s посредством системы функций $\langle \min, \max \rangle$, суть которого состоит в перестановке всех неупорядоченных пар по кольцу КВС. В случае упорядоченности массива применение оператора TRANSP-s не изменяет содержимое кольца.

На КВС определён также оператор $P \succ$ сдвига содержимого кольца на одну позицию по часовой стрелке, в результате которого изменяется содержимое пар. Кроме того, на КВС зададим предикат Psort – истинный, когда массив остаётся упорядоченным до и после очередного сдвига.

Алгоритм SIN_SORT синхронной сортировки массива на кольце КВС представим следующей циклической параллельной регулярной схемой

$$\text{SIN_SORT} ::= \{ [\text{Psort}] \text{TRANSP-s} * P \succ \}.$$

Алгоритм SIN_SORT состоит в циклической сортировке массива в результате синхронной перестановки всех неупорядоченных пар посредством неоднородного ПО-преобразования TRANSP-s (определённого выше) с последующим сдвигом $P \succ$. Проверка сортируемого массива на упорядоченность осуществляется посредством предиката Psort – истинного в случае неизменности массива при двух последовательных выполнениях тела цикла.

Отметим, что приведенная регулярная схема является аналогом шаблона в объектно-ориентированных средах. Она задает стратегию обработки (сортировки), которая применима к данным различных типов. Данная схема может быть представлена в виде различных изобразительных средств – регулярных схем, САА-схем с развернутым комментарием к каждому оператору и предикату, граф-схем, по которым с помощью инструментария производится автоматизированное диалоговое проектирование программного кода в объектно-ориентированный язык программирования с использованием техники шаблонов.

Пример 2. Пусть дана линейная вычислительная структура ЛВС, ориентированная на ПО-преобразование данных с переменной окрестностью. Содержимым структуры ЛВС служит числовой размеченный массив M_0 с указателями, фиксирующими пары элементов, подлежащих перестановкам. Следует отметить, что в отличие от алгоритма SIN_SORT (см. пример 1) обработка осуществляется с переменным шагом-окрестностью, определяющей доступ к обрабатываемым данным. Рассмотрим известный метод сортировки массивов, принадлежащий Шеллу [16, 18]. Метод Шелла (МШ) весьма эффективный метод последовательной сортировки. В отличие от класса алгоритмов сортировки, базирующихся на перестановках соседних элементов (к данному классу относится и приведенный выше алгоритм SIN_SORT), в основу метода Шелла положена перестановка элементов, расположенных на расстоянии $k > 0$ друг от друга [18]. Данный метод представим следующей РС:

$$\text{МШ} ::= \text{ИНИЦ} * \{ [p] \text{РУ}(k) * \text{СОРТ}(k) * \text{УСТ}(H) \} * \text{ФИН}, \quad (5)$$

где ИНИЦ – оператор инициализации с выбором начального значения параметра $k := m$ (в частности $k := n$, где n – длина массива);

p – предикат истинный в случае упорядоченности сортируемого массива;

$\text{РУ}(k) ::= \{ [d(Y_1, Y_2) = m] \Pi(Y_1) \}$ – оператор рассредоточения указателей на расстояние в m элементов,

$m = \lfloor k/2 \rfloor$ – целая часть от деления;

$\Pi(Y_1)$ – сдвиг указателя Y_1 на символ вправо;

$\text{СОРТ}(k)$ – регулярная схема РС алгоритмов сортировки подмассивов

$$m_i : H \ Y_1 \ Y_2 \ a_i \ a_{i+k} \ a_{i+2k} \ \dots \ a_{i+mk} \ \dots \ a_n \ K \quad (6)$$

($i = 1, 2, \dots, k$) (предполагается, что $H < a_i < K$ для любого элемента a_i сортируемого массива и при достижении маркера K указатели далее не перемещаются);

$\text{УСТ}(H)$ – оператор установки Y_1 и Y_2 на маркер H – начало массива.

Следует отметить, что РС СОРТ(k) может быть проинтерпретирована посредством подходящих алгоритмов, рассмотренных в [19-21].

Таким образом, в соответствии с МШ сортируемый массив представляет собой k подмассивов, допускающих асинхронную мультиобработку различными алгоритмами сортировки. Каждый из таких алгоритмов может быть сориентирован, например, на синхронную обработку своего подмассива, подобно алгоритму SIN_SORT (см. пример 1).

3.4. Функциональные построения клонов n -арных отношений. Рассмотрим алгебру функциональных отношений, сигнатура которой включает бинарную операцию, подобную свёртке де Моргана (см. разд.1),

которую назовём суженной свёрткой функциональных n -арных отношений. Пусть $F^c \subset M_1 * M_2 * \dots * M_c$ и $G^q \subset M_1 * M_2 * \dots * M_q$ суть функциональные n -арные отношения, определённые на декартовых произведениях указанных множеств, где $M_c, M_q = 1, 2, \dots$

Суженная свёртка определяется следующим образом: результирующее $((c+q)-1)$ -арное отношение $H^{(k+c)-1} = F^c * G^q$ представляет собой совокупность кортежей элементов

$$w_{k+c} = \{ \tilde{z} \mid \tilde{z} \in M_1 * M_2 * \dots * M_{c+q} \}, \text{ здесь } \tilde{z} = (a_1, a_2, \dots, a_c, a_{c+1}, \dots, a_{c+q}), \text{ причём существуют } c$$

промежуточных элементов b_1, b_2, \dots, b_c таких, что

$$(a_1, a_2, \dots, a_{q-1}, b_1) \in G^q, (a_2, a_3, \dots, a_q, b_2) \in G^q, \dots, (a_c, a_{c+1}, \dots, a_{(c+q)-1}, b_c) \in G^q,$$

для которых выполняется следующая суперпозиция n -арных отношений

$$H = F * G(x_1, x_2, \dots, x_{c+q}) = F(G^q, G^q, \dots, G^q), \quad (7)$$

причём кортеж $(a_1, a_2, \dots, a_c, a_{c+1}, \dots, a_{c+q}) \in F * G$ и суженная суперпозиция функциональных n -арных отношений сопряжена с суперпозицией (4) функций, ассоциированных с n -отношениями F и G .

Отметим, что n -арное отношение $F * G$ суть последовательное вычисление n -арных отношений G^q , а затем c -арного отношения F в соответствии с суперпозицией (7).

Рассмотрим, в частности, множества MASS всех неупорядоченных числовых массивов и MUM их отсортированных образов так, что пара $(M, M') \in SORT$, если $M \in MASS$, $M' \in MUM$ и M' получен из M в результате перестановки его элементов; тем самым бинарное отношение SORT функционально и служит постановкой задачи последовательной или параллельной сортировки в зависимости от выбора алгоритма и соответствующей вычислительной среды (см. 3.3, примеры 1, 2).

Пусть Z – система образующих (CO), состоящая из операций над n -отношениями, рассмотренных в [10] и порождающая базовые конструкции алгебры Кодда [12, 13]. Замкнём Z по обобщению свёртки де Моргана на n -местный случай (см. разд. 1, теорема 1, следствия).

Таким образом, из проведенных построений вытекает справедливость следующего утверждения.

Теорема 3.

а) суженная свёртка $F * G$ функциональных n -арных отношений F и G порождает новое функциональное отношение согласно суперпозиции (4);

б) бинарное отношение SORT функционально и служит постановкой задачи на последовательную или параллельную сортировку в зависимости от выбора алгоритма и соответствующей вычислительной среды (см. разд. 3.3, примеры 1, 2);

в) для теории клонов и решёток их подалгебр, исследованных в [11, 22, 23], могут быть построены ассоциированные с ними клоны функциональных n -отношений, на которые распространяются соответствующие теоремы о функциональной полноте;

г) замыкая рассмотренные в [9, 12, 13] системы образующих из n -отношений по приведенным в данной статье вариантам их суперпозиций, построим клоны, порождающие семейства алгебр функциональных n -отношений, адекватных по порождающей мощности системам образующих алгебр Кодда [12, 13].

3.5. Представленный в данной статье математический аппарат позволяет исследовать алгоритмическую структуру знаний предметных областей и тем самым определять важные для данной предметной области алгоритмические зависимости, порождать новые алгоритмические знания, устанавливать взаимосвязи между близкими предметными областями. В то же время он положен в основу инструментальных средств создания программных систем. Аппарат алгебры алгоритмики предлагает высокоуровневые средства проектирования алгоритмов и структур данных специалистам предметных областей, не владеющих в совершенстве техникой объектно-ориентированного программирования. Эти средства составляют основу восходящего от синтезатора МУЛЬТИПРОЦЕССИСТ [24] интегрированного инструментария проектирования и синтеза алгоритмов и программ для погружения в объектно-ориентированные среды с подключением развитых в них инструментальных средств [25-29]. Например, приведенная в статье схема сортировки массивов задает только алгоритмическую стратегию обработки и взаимодействия операторов, предикатов с данными сортируемого массива через технику указателей, а по ней синтезируется шаблон класса реализации такого алгоритма сортировки. Таким образом, дальнейшее развитие алгебры алгоритмики в плане построения клонов n -отношений и соответствующих алгебр, ориентированных на формализацию и проектирование структур данных для различных предметных областей послужит мощным стимулом для дальнейшего развития исследований в теоретическом, системном и прикладном программировании.

Сформулируем еще одну важную задачу, решение которой в плане построения и исследования алгоритмической структуры предметной области позволит синтезировать и сопровождать программные системы реше-

ния экономических задач. Рассмотрим в качестве иллюстративного примера такого формализованного проектирования с использованием клонирования n -отношений задачу первичного структурированного накопления информации в экономических задачах как результат осуществления хозяйственных или другого рода операций. Предложенную ниже модель будем называть учетной. Предполагается, что прикладной пользователь имеет возможность легко модифицировать эту модель с целью настройки ее для решения задач учета конкретной предметной области (бухгалтерского, финансового, управленческого, статистического и других видов учета для объектов различной природы). Основными элементами такой модели являются счета, аналитические объекты, журнал регистрации операций, документы для регистрации операций, отчеты. Счета представляют собой некоторые учетные регистры, на которых накапливаются итоговые данные по осуществленным операциям, эти итоги формируются с учетом временного интервала проведения операции. Такие итоговые данные могут быть разбиты на составляющие в соответствии с объектами, принимающими участие в осуществлении операции. Например, счет "товар" может содержать сумму имеющихся на складе товаров и в то же время эта сумма может быть разбита на составляющие суммы по каждому товару. В данном случае говорят о наличии аналитического учета по данному счету, а сами объекты, по которым идет разбиение итоговых данных, – аналитическими. Каждую операцию при функционировании предложенной модели регистрируют посредством соответствующего ей документа в специальном журнале операций. Документы представляют собой алгоритмы формирования итоговых данных на счетах, задействованных в данной операции. Для документов пользователь самостоятельно может определить структуру входной информации, характеризующую операцию (обычно в виде табличной ведомости), указать арифметические формулы расчетов для получения по исходной информации итоговых регистрационных данных (аналогично электронным таблицам), описать алгоритмическую схему формирования итогов на счетах и создать печатную форму документа регистрируемой операции при необходимости. Отчеты представляют собой отображение итоговых данных по различным временным интервалам, разбиение итоговых данных по аналитическим объектам, по операциям, по взаимодействию между отдельными аналитическими объектами, отчеты используются не только для анализа итоговых данных, но и для редактирования процесса регистрации операций. Исходя из сказанного основу модели составляют следующие n отношения:

R_1 – план счетов, отношение определено на декартовом произведении следующих множеств: <номера счетов>*<наименования счетов>*<группы аналитических объектов, связанных с данным счетом>. Множество <группы аналитических объектов> может быть пустым, в этом случае для выбранного счета не ведется аналитический учет.

R_2 – группы аналитических объектов (назовем их справочниками), отношение определено на декартовом произведении множеств <номера справочников>*<наименования справочников>*<атрибуты (свойства) аналитического объекта>, в данном случае <атрибуты аналитического объекта> могут быть множеством или отношением, если количество атрибутов данного объекта больше 1 (например, аналитический объект "организация" характеризуется набором атрибутов, которые могут быть описаны в виде следующего отношения: <наименование организации>*<юридический адрес>*<банковский расчетный счет> и т.д.).

R_3 – журнал регистрации учетных операций, отношение определено на следующем декартовом произведении: <дата операции>*<наименование документа, регистрирующего операцию>*<состояние операции – осуществленная или планируемая>.

R_4 – документы регистрации операций: отношение определено следующим образом: <номер документа>*<структура входной информации для документа>*<программа формирования накопления итогов на счетах, затрагиваемых данной операцией>*<печатная форма отчета по данной операции>. Здесь <структура входной информации для документа> представляет собой отношение, описывающее в виде таблицы структуру входной информации, которая должна быть заполнена и рассчитана во время регистрации операции. Форма отчета по операции также определяется в виде отношения, тогда как программа формирования итоговых результатов по факту регистрации данной операции может быть описана САА-схемой алгоритма ее функционирования над входными данными документа и итоговыми данными учетной модели. Перечисленные отношения определяют структуру данных функционирования такой модели, а алгоритмы регистрации и накопления информации описываются в виде САА-схем, по которым с помощью инструментальных средств строятся программы в выбранном языке программирования. Инструментальные средства алгоритмики кроме автоматизированных средств построения программ по их описаниям в САА-схемах содержат средства преобразования n -отношений описания данных в предметной области в структуру той базы данных, в среде которой будет осуществляться проектирование программной системы функционирования такой модели. Манипулирование данными, представленными в виде n -отношений, сводится к использованию перечисленных (разделы 1-4) операций в клоне n -отношений и автоматическому синтезу по выражениям в такой алгебре соответствующих операторов языка SQL.

Заключение

Выполненное исследование относится к интенсивно развиваемой на Западе области знаний, названной алгебраическая алгоритмика [30]. Для данной области характерно гармоничное соединение современной алгеб-

ры и информатики. Точнее, алгебраические результаты, относящиеся, в частности, к различным алгебрам (полугруппы, кольца, системы уравнений и пр.), дополняются алгоритмами и программами обработки соответствующих структур данных. При этом вне поля зрения осталась алгебра алгоритмов [8], находящаяся на стыке алгебры и информатики.

В данной статье получены следующие результаты:

- построены клоны n -отношений для создания алгебраического аппарата, нацеленного на совместное проектирование структур данных различной природы и алгоритмов их обработки;
- намечен подход к распространению результатов по теории полугрупповых клонов (клоны формальных языков и грамматик, автоматные и алгоритмические клоны [22, 23]) на построенные в данной статье клоны n -отношений;
- на базе полученных результатов открывается возможность дальнейшего развития интегрированного инструментария алгебры алгоритмики с целью обеспечения разноаспектного проектирования и синтеза программ по известной формуле Вирта: алгоритмы + структуры данных = программы.

В качестве перспективы дальнейшего развития аппарата алгебры алгоритмики сформулируем ряд нерешённых задач для последующих исследований по алгебре алгоритмики, которая восходит от результатов и парадигм В.М. Глушкова [31].

1. Построение системы образующих для клона алгебр n -отношений, порождающей семейство алгебр, включающее алгебру Кодда.

2. Решение проблемы функциональной полноты для построенного клона и исследование свойств решётки его подалгебр, а также поверхности – фрагмента, состоящего из конечнопорождённых подалгебр данного клона.

3. Разработка на базе клонов алгебр n -отношений инструментальных средств для совместного проектирования структур: данных, операторов и предикатов для решения прикладных задач, относящихся к различным предметным областям.

4. Усовершенствование на базе аппарата алгебр n -отношений методов интенсивного обучения аудитории (в том числе и виртуальной) для проектирования и автоматизации построения обучающих систем, включая и дистанционные.

5. Озвучивание созданных инструментальных средств, с ориентацией их на обучение пользователей, имеющих физические ограничения, включая и лиц с проблемами зрения.

1. Кон П. Универсальная алгебра. – М.: Мир, 1968. – 348 с.
2. Курош А.Г. Лекции по общей алгебре. – М.: Физматгиз. – 1962. – 396с.
3. Post E.L. The two-valued iterative systems of mathematical logic // Ann. Math. Studies. – 1941. – Vol. 5. – P. 147.
4. Яблонский С.В. Функциональные построения в k -значной логике // Труды МИАН СССР. – 1958. – Т.51. – С. 5-142.
5. Мальцев А.И. Итеративные алгебры и многообразия Поста // Избр. тр. А.И.Мальцева. – 1976. – Т.2. – С. 316-330.
6. Цейтлин Г.Е. Алгебраическая алгоритмика: теория и приложения // Кибернетика и системный анализ. – 2003. – №1. – С. 8-18.
7. Цейтлин Г.Е. Алгебры Глушкова и теория клонов // Кибернетика и системный анализ. – 2003. – №4. – С. 48-58.
8. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра. Языки. Программирование. – К.: Наук. думка. – 1-е изд., 1974. – 327с.; 2-е изд., перераб., 1978. – 318 с.; 3-е изд., перераб. и доп., 1989. – 376 с. Gluschkow W.M., Zeitlin G.E., Justchenko J.L. Algebra. Sprachen. Programmierung. – Berlin: Akademie-Verlag, 1980. – 340 p.
9. Цейтлин Г.Е. О бесконечно порождённых подалгебрах модифицированной алгебры Поста // Кибернетика. – 1971. – №2. – С. 43-56.
10. Теория Галуа для алгебр Поста. – Ч. 1, 2 / В.Г. Бондарчук, Л.А. Калужнин, В.Н. Котов, Б.А. Ромов // Кибернетика. – 1969. – №3. – С. 1-10; №5. – С. 1-9.
11. Цейтлин Г.Е. Введение в алгоритмику. – К.: Сфера, 1998. – 310 с.
12. Codd E.F. A Relational Model of Data for Large Shared Data Banks // Comm. ACM. – 1970. – Vol. 13. – №6. – P. 377-387.
13. Дейт К.Дж. Введение в системы баз данных. – КГ: Диалектика, 1999. – 784 с.
14. Tsejtin G.E. The theory of modified Post algebras and multidimensional automata structures. – Lecture Note in Computer Science. – 1975. – №32. – P. 418-424.
15. Бондарчук В.Г., Цейтлин Г.Е. Об алгебрах периодически определенных преобразований бесконечного регистра // Кибернетика. – 1969. – №1. – С. 18-28.
16. Кнут Д. Искусство программирования для ЭВМ. – М.: Мир, 1978. – Т. 3. – 843 с.
17. Лорин Г. Сортировка и системы сортировки. – М.: Наука, 1983. – 384 с.
18. Vladimir Estivill-Castro and Derick Wood. A Survey of Adaptive Sorting Algorithms. // ACM Computing Surveys, 1992. – Vol. 24. – №4, December. – P. 441-476.
19. Цейтлин Г.Е. Формальная трансформация структурированных алгоритмов сортировки // Программирование. – 1985. – №2. – С. 79-91.
20. Цейтлин Г.Е. Проектирование алгоритмов параллельной сортировки // Программирование. – 1989. – №6. – С. 4-19.
21. Цейтлин Г.Е. Поиск и сортировка: классификация, трансформация, синтез. I, II // Автоматика и телемеханика. – 1992. – №4. – С. 147-154; №5. – С. 156-165.
22. Цейтлин Г. Е. Проблема функциональной полноты для мета-алгебр регулярных событий // Кибернетика и системный анализ. – 2000. – №6. – С. 14-27.
23. Цейтлин Г. Е. Проблема функциональной полноты в итеративных мета-алгебрах // Кибернетика и системный анализ. – 1998. – №2. – С. 28-45.
24. Многоуровневое структурное проектирование программ: Теоретические основы, инструментарий / Е.Л. Ющенко, Г.Е. Цейтлин, В.П. Грицай, Т.К. Терзян. – М.: Финансы и статистика, 1989. – 208 с.
25. Интегрированный инструментарий проектирования и синтеза классов алгоритмов и программ / Цейтлин Г.Е., Амонс А.А., Головин О.В., Зубов А.Ю. - Кибернетика и системный анализ. – 2000. – №3. – С. 165-170.
26. Цейтлин Г.Е., Теленик С.Ф., Амонс А.А. Алгебро-логическая формализация в объектно-ориентированных технологиях // Пробл. программирования. – 2002. – № 1-2. – С. 136-146.

27. *Цейтлин Г.Е., Яценко Е.А.* Элементы алгебраической алгоритмики и объектно-ориентированный синтез параллельных программ // Математические машины и системы. – 2003. – № 2. – С. 64-76.
28. *Яценко Е.А., Мохница А.С.* Инструментальные средства конструирования синтаксически правильных параллельных алгоритмов и программ // Пробл. программирования. Спец. выпуск по материалам 4-й Междунар. науч.-практ. конф. по программированию УкрПРОГ'2004. – 2004. – №2-3. – С. 444-450.
29. *Яценко Е.А.* Алгебры гиперсхем и интегрированный инструментальный синтеза программ в современных объектно-ориентированных средах. // Кибернетика и системный анализ. – 2004. – №1. – С. 47-52.
30. *Ноден П., Китте К.* Алгебраическая алгоритмика (с упражнениями и решениями). – М.: Мир, 1999. – 720 с.
31. *Капитонова Ю.В., Лещевский А.А.* Парадигмы и идеи академика В.М.Глушкова. – К: Наук. думка, 2003. – 330 с.