УДК 519.681

В.Г. Акуловский, А.Е. Дорошенко, Е.А. Яценко

ПОЛНОТА АЛГЕБРЫ АЛГОРИТМОВ С ДАННЫМИ

Показана полнота алгебры алгоритмов с данными, предназначенной для согласованного проектирования потоков управления и обрабатываемых данных. Рассматривается также реализуемость разработки алгоритмов от управления и от данных с использованием как восходящей, так и нисходящей стратегий проектирования.

Ключевые слова: алгебра алгоритмов с данными, полнота алгебраического аппарата, восходящая и нисходящая стратегии проектирования, подходы к разработке от управления и от данных.

Введение

О чрезвычайной важности роли данных в процессе разработки алгоритмов и программ и, соответственно, о неразрывной связи программ и обрабатываемых данных и невозможности рассматривать одни в отрыве от других известно достаточно давно [1, 2]. Вместе с тем, задача совместного и согласованного проектирования потоков управления и данных в рамках существующих формальных спецификаций алгоритмов не рассматривается (см. например, [3]). Для решения этой задачи был разработан алгебраический аппарат (алгебра алгоритмов с данными) [4].

К числу важных алгебраических проблем относится доказательство полноты алгебраического аппарата. В процессе рассмотрения этой проблемы будем учитывать, что проектирование алгоритма, как известно, может осуществляться в рамках восходящей и нисходящей стратегий разработки, с использованием подходов к программированию – от управления (функционально-ориентированный подход) и от данных.

Цель данной работы – доказательство возможности разработки средствами предложенной алгебры произвольных алгоритмов с согласованными описаниями потоков управления и обрабатываемых данных на каждом шаге проектирования, при реализации указанных стратегий и подходов к программированию.

1. Алгебра алгоритмов с данными

Алгебра алгоритмов с данными построена в результате модификации известной модели ЭВМ Глушкова [5], которая расширена за счет введения носителя данных. Носитель данных отождествляется с памятью и устройствами ввода-вывода, а множество хранимых данных обозначено D^{AC}. Данными называется пара

$$D = \left\langle N_j, З \right\rangle \ (D \in D^{\mathrm{AC}}),$$

где $З = \left\langle з_1,..., з_n \right\rangle$ – кортеж последовательно расположенных значений, определяющий в каждый момент времени текущее состояние данных, носитель которого N_j, однозначно идентифицируется индексом $j \in J$, который интерпретируется как адрес, принадлежащий множеству доступных адресов J.

Введены Д-операторы $(D)X(D')$, на входе и выходе которых специфицированы множества обрабатываемых этим Д-оператором данных. Приведем определения Д-операторов, актуальных для данной работы.

Определение 1. Д-оператор $(D)O(D')$ обрабатывает множество данных $D^o \subseteq D^{\mathrm{AC}}$ и только это множество. Процесс обработки представляет собой некоторую конечную последовательность элементарных действий (операций), которую будем называть функциональностью этого Д-оператора. Функциональность $(D)O(D')$ и множество данных D^o "согласованы", если эта функциональность необходима и достаточна для обработки множества данных D^o, а это множество необходимо и достаточно для реализации этой функциональности. Входные D и, изменяемые в результате выполнения Д-оператора, выходные D' данные называются наблюдаемыми, а данные, образующие множество $D^{nn} = D^{o} \setminus (D \cup D')$, – ненаблюдаемым.

Предложенное определение позволяет на каждом уровне детальности представления Д-оператора рассматривать только актуальные (наблюдаемые) для данного уровня данные D и D' и абстрагироваться от неактуальных (ненаблюдаемых) данных D^{nn}.

Второй тип Д-оператора – $(D^{\pi})P(\alpha)$, называемый предикатом, представляет собой n-местную (в общем случае) логическую функцию $P: D^{\pi} \to \alpha \in E_2$, где D^{π} – график некоторого n-арного отношения, $E_2 = \{1, 0\}$, α – двузначное логическое условие, анализирующее множество данных D^{π} и продуцирующее логическое условие α, характеризующее отношение между текущими значениями этих данных.

На множестве логических условий определены следующее известные булевы операции: конъюнкция ($\wedge$), дизъюнкция ($\vee$), отрицание ($\overline{\alpha}$) с известными таблицами истинности (см. например, [6]).

На множестве Д-операторов определены операции, из числа которых приведем следующие.

Композиция Д-операторов $(D)O(D') * (\widetilde{D})\widetilde{O}(\widetilde{D}')$ означает последовательное выполнение сначала Д-оператора $(D)O(D')$, а затем Д-оператора $(\widetilde{D})\widetilde{O}(\widetilde{D}')$. Композиция предикатов, с учетом введенных логических операций, записывается в виде $(\dot{D}^{\pi})\dot{P}(\alpha) \circ (\ddot{D}^{\pi})\ddot{P}(\alpha)$, где $\circ$ – одна из логических операций.

р-дизъюнкция

$$[(D^{\pi})P(\alpha)]((D_1)O_1(D_1') * (D_2)O_2(D_2')) = $$

$$= \begin{cases} (D_1)O_1(D_1'), \text{если } \alpha = 1; \\ (D_2)O_2(D_2'), \text{если } \alpha = 0, \end{cases}$$

в результате выполнения которой выбирается один из двух возможных Д-операторов в соответствии со значением логического условия α.

р-итерация $[(D^{\pi})P(\alpha)]\{(D)O(D')\}$ осуществляет циклическое выполнение Д-оператора $(D)O(D')$ при $\alpha = 1$ и завершается в противном случае.

Исходя из того, что элементарные данные $D^{Э}$ – это данные, не подлежащие детализации, приведем определение иерархически организованных (в дальнейшем, иерархических) данных.

Определение 2. Любые данные $D^{И}$ – иерархические ($D^{И} \in S^{И}$, где $S^{И}$ – множество иерархических данных), если они представляют собой иерархическое множество

$$D^{И} = \left\{ D_{i_1, i_2, \ldots, i_k} \mid i_k (k = 1, 2, \ldots, n) \right\},$$

где k – уровень иерархии, n – число уровней иерархии, i_k – номер элемента на k-м уровне иерархии ($i_1 = 1$). Каждый j-й уровень иерархии представляет собой множество данных $\{D_{1_j}, D_{2_j}, \ldots, D_{p_j}\}$, реализующих более детальное представление данных предыдущего ($j-1$)-го уровня. Каждая i-я спецификация j-го уровня D_{i_j} детализуется и на следующем ($j+1$) уровне представляет множество $\{D_{1_{j+1}}, D_{2_{j+1}}, \ldots, D_{p_{j+1}}\}$. Последний ($k$-й) уровень иерархии представляет собой множество элементарных данных $S'^{Э} = \{D^{э}_{k_1}, \ldots, D^{э}_{k_p}\}$ такое, что $S'^{Э} \subseteq S^{Э}$, где $S^{Э} \subset S^{И}$ – конечное множество элементарных данных.

На множестве иерархических данных введены следующие операции.

Операция **укрупнения** $*\{D_1, D_2, \ldots, D_k\} = D$ определена на множестве $S^{И}$. Результатом выполнения этой операции являются данные на более высоком уровне иерархии.

Операция **детализации** данных $\overline{*}\, D = \{D_1, D_2, \ldots, D_k\}$ определена на множестве $S^{и}$ и не определена на множестве $S^{Э}$. Результатом выполнения этой операции являются данные на более низком уровне иерархии.

Таким образом, предложена система алгоритмических алгебр

$$\text{САА} \setminus \text{Д} ::= \left\langle \{X, L, S^{и}\}; \Omega \right\rangle,$$

где X – множество Д-операторов, L – множество логических условий, $S^{и}$ – множество иерархических данных, Ω – сигнатура алгебры, включающая операции Ω_1, определенные на множестве X, операции Ω_2, определенные на множестве L, и Ω_3, определенные на этом множестве $S^{и}$.

Проблему полноты алгебраической системы начнем с рассмотрения возможностей декомпозиции и объединения Д-операторов.

2. Объединение и декомпозиция Д-операторов

Воспользовавшись результатами, полученными в работах [7, 8], докажем, что Д-операторы могут быть объединены и декомпозированы. Прежде чем перейти к доказательству, введем некоторые новые понятия и термины, необходимые для его осуществления.

Для описания алгоритмов используются регулярные схемы (РС), представляющие собой суперпозиции Д-операторов и операций алгебры [5, 6]. Учитывая, что на функциональность Д-операторов не накладываются ограничения, будем рассматривать операции алгебры как специальный тип Д-операторов. То есть операции р-дизъюнкции

$$[(D^{\pi})P(\alpha)]((D_1)O_1(D_1') * (D_2)O_2(D_2'))$$

соответствует Д-оператор

$$(D)O^{д}(D'), \text{ где } D = (D^{\pi} \cup D_1 \cup D_2), \quad D' = (D_1' \cup D_2') \qquad (1)$$

а р-итерации $[(D^{\pi})P(\alpha)]\{(D)O(D')\}$ – Д-оператор

$$(D)O^{и}(D'), \text{ где } D = (D^{\pi} \cup D), \quad D' = \dot{D}'. \qquad (2)$$

При записи алгоритмов в виде суперпозиции Д-операторов, включающих введенные выше, полученные схемы будем называть композиционными (КС). В общем случае будем говорить, что алгоритм описывается с помощью схемы алгоритма (СА).

Понятие, – состояние вычислительного процесса (ВП), введенное в работе [4], определим, учитывая специфику данной работы, в упрощенном варианте.

Определение 3. Исходным для Д-оператора $(D)O(D')$ состоянием ВП называется множество данных $\hat{D}^T$, где $D \subseteq \hat{D}^T$, значения которых изменяются в результате и только в результате выполнением Д-оператора, в результате чего ВП переходит в результирующе состояние $\breve{D}^T$, где $D' \subseteq \breve{D}^T$. Композиция Д-операторов $(\dot{D})\dot{O}(\dot{D}') * (\tilde{D})\tilde{O}(\tilde{D}')$ переводит ВП из исходного состояния $\hat{D}^T$, где $\dot{D} \subseteq \hat{D}^T$, в состояние $\hat{\breve{D}}^T$, где $\dot{D}', \tilde{D} \subseteq \hat{\breve{D}}^T$, и состояние $\breve{D}^T$, где $\tilde{D}' \subseteq \breve{D}^T$. Для любого состояния ВП D^T выполняется $D^{\text{AC}} \subseteq D^T$.

Теперь остановимся на рассмотрении особенностей выполнения композиции Д-операторов $(\dot{D})\dot{O}(\dot{D}') * (\tilde{D})\tilde{O}(\tilde{D}')$. Каждый из Д-операторов, входящий в композицию, решает свою локальную задачу по обработке локальных данных. При этом композиция Д-операторов решает некоторую глобальную задачу по преобразованию глобальных входных данных в выходные, характерную для данной композиции. Решение этой глобальной задачи осуществляется как независимо каждым из Д-операторов, так и совместно обоими Д-операторами.

В связи с указанными особенностями выполнения композиции Д-операторов, введем следующее определение.

Определение 4. В композиции Д-операторов $(\dot{D})\dot{O}(\dot{D}') * (\tilde{D})\tilde{O}(\tilde{D}')$ на входе и выходе $(\dot{D})\dot{O}(\dot{D}')$ специфицированы

$\dot{D} = \dot{D}'' \cup \dot{D}^{\text{Л}}$ и $\dot{D}' = \dot{D}^{\text{СБ}} \cup \dot{D}'^{\text{Л}} \cup D^{\text{СВ}}$, а на входе и выходе $(\widetilde{D})\widetilde{\text{O}}(\widetilde{D}')$ – $\widetilde{D} = \widetilde{D}^{\text{СБ}} \cup \widetilde{D}^{\text{Л}} \cup D^{\text{СВ}}$ и $\widetilde{D}' = \widetilde{D}''' \cup \widetilde{D}'^{\text{Л}}$, где $\dot{D}''$, $\widetilde{D}'''$, $\dot{D}^{\text{СБ}}$, $\widetilde{D}^{\text{СБ}}$ – глобальные, $\dot{D}^{\text{Л}}$, $\dot{D}'^{\text{Л}}$, $\widetilde{D}^{\text{Л}}$, $\widetilde{D}'^{\text{Л}}$ – локальные данные. $\dot{D}^{\text{СБ}}$ и $\widetilde{D}^{\text{СБ}}$ – собственные данные такие, что $\widetilde{D}^{\text{СБ}}$ не обрабатываются Д-оператором $(\dot{D})\dot{O}(\dot{D}')$, а $\dot{D}^{\text{СБ}}$ – $(\widetilde{D})\widetilde{\text{O}}(\widetilde{D}')$. $D^{\text{СВ}}$ – связывающие данные такие, что $D^{\text{СВ}} = \dot{D}' \cap \widetilde{D}$. Локальные данные обрабатываются только одним Д-оператором и специфицированы только на его входе и выходе.

Определение 4 поясним следующим образом.

Собственные данные $\dot{D}^{\text{СБ}}$ являются результатом выполнения только Д-оператора $(\dot{D})\dot{X}(\dot{D}')$, $\widetilde{D}^{\text{СБ}}$– обрабатываются только Д-оператором $(\widetilde{D})\widetilde{\text{X}}(\widetilde{D}')$, а связывающие данные $D^{\text{СВ}}$ обрабатываются совместно двумя Д-операторами. В частных случаях, любое из множеств данных на входе и выходе Д-операторов, но не все сразу, могут быть пустыми.

Теперь определим понятие эквивалентности Д-оператора и композиций Д-операторов.

Определение 5. Д-оператор $(D)O(D')$ и композиция Д-операторов $(\dot{D})\dot{O}(\dot{D}') * *(\widetilde{D})\widetilde{\text{O}}(\widetilde{D}')$ эквивалентны, если, при равенстве их исходных состояний ВП ($D^T = \widehat{D}^T$), результатом их выполнения явятся равные результирующие состояния ($D'^T = \breve{D}^T$). Д-оператор $(D)O(D')$ принадлежит к более высокому (укрупненному), а композиция к более низкому (детальному) уровням описания алгоритма, если на входе и выходе $(D)O(D')$ специфицированы только глобальные данные эквивалентной ему композиции.

Доказательство возможности объединения Д-операторов предварим следующей леммой.

Лемма 1. В композиции Д-операторов $(\dot{D})\dot{O}(\dot{D}') * (\widetilde{D})\widetilde{\text{O}}(\widetilde{D}')$ для собственных данные $\dot{D}^{\text{СБ}}$ и $\widetilde{D}^{\text{СБ}}$ выполняются соотношения $\dot{D}^{\text{СБ}} \subseteq \breve{D}^T$, $\widetilde{D}^{\text{СБ}} \subseteq \widehat{D}^T$, а ненаблюдаемые данные D^{nn} входят в состав данных, образующих состояние ВП.

Доказательство. Из определений 1, 3, 4 известно, что $\widetilde{D}^{\text{СБ}}, \dot{D}^{\text{СБ}}, D^{nn} \subset D^{\text{АС}}$, а для собственных данных выполняются соотношения $\widetilde{D}^{\text{СБ}}, \dot{D}^{\text{СБ}} \subseteq \widehat{\breve{D}}^T$.

Поскольку, в соответствии с определением 3, для любого D^T выполняется $D^{\text{АС}} \subseteq D^T$, то ненаблюдаемые данные D^{nn} входят в состав данных, образующих состояния ВП ($D^{nn} \subset D^T$), а для $\widetilde{D}^{\text{СБ}}, \dot{D}^{\text{СБ}}$, которые, в соответствии с определением 4, не изменяются Д-операторами $(\dot{D})\dot{O}(\dot{D}')$ и $(\widetilde{D})\widetilde{\text{O}}(\widetilde{D}')$, выполняются соотношения $\dot{D}^{\text{СБ}} \subseteq \breve{D}^T$, $\widetilde{D}^{\text{СБ}} \subseteq \widehat{D}^T$.

Лемма доказана.

Покажем, что любая композиция Д-операторов может быть преобразована в новый Д-оператор.

Теорема 1. В результате объединения произвольных Д-операторов (в частности, $(D)O^{\text{Д}}(D')$, $(D)O^{\text{И}}(D')$), входящих в композицию $(\dot{D})\dot{O}(\dot{D}') * (\widetilde{D})\widetilde{\text{O}}(\widetilde{D}')$, будет осуществлен переход на более высокий (укрупненный) уровень представления алгоритма. Полученный в результате объединения Д-оператор $(D)O(D')$ будет эквивалентен исходной композиции, если его входные и выходные данные такие, что $D = (\dot{D}'' \cup \widetilde{D}^{\text{СБ}})$, где $\dot{D}'' = (\dot{D} \setminus \dot{D}^{\text{Л}})$, и $D' = (\dot{D}^{\text{СБ}} \cup \widetilde{D}''')$, где $\widetilde{D}''' = (\widetilde{D}' \setminus \widetilde{D}^{\text{Л}})$. При этом, если функциональность каждого из исходных Д-операторов задана, а обрабатываемые данные с ней согласованы, то у производного Д-оператора функциональность то же будет задана, а обрабатываемые им данные с ней согласованы.

Доказательство. Запишем данные в композиции Д-операторов $(\dot{D})\dot{O}(\dot{D}') * *(\widetilde{D})\widetilde{\text{O}}(\widetilde{D}')$, в соответствии с определением 4. В результате получаем её в виде

$$(\dot{D}'', \dot{D}^{\text{Л}})\dot{O}(\dot{D}^{\text{СБ}}, \dot{D}'^{\text{Л}}, D^{\text{СВ}}) *$$
$$* (D^{\text{СВ}}, \tilde{D}^{\text{СБ}}, \tilde{D}^{\text{Л}})\tilde{O}(\tilde{D}''', \tilde{D}'^{\text{Л}}).$$

Собственные данные $\dot{D}^{\text{СБ}}$ и $\tilde{D}^{\text{СБ}}$ переместим на вход $\dot{O}$ и выход $\tilde{O}$, соответственно. В результате этого, получаем композицию Д-операторов

$$(\tilde{D}^{\text{СБ}}, \dot{D}'', \dot{D}^{\text{Л}})\dot{O}(\dot{D}'^{\text{Л}}, D^{\text{СВ}}) *$$
$$* (D^{\text{СВ}}, \tilde{D}^{\text{Л}})\tilde{O}(\tilde{D}''', \tilde{D}'^{\text{Л}}, \dot{D}^{\text{СБ}}).$$

Локальные и связывающие данные переведем в разряд ненаблюдаемых ($\dot{D}^{\text{Л}}, \dot{D}'^{\text{Л}}, D^{\text{СВ}}, \tilde{D}^{\text{Л}}, \tilde{D}'^{\text{Л}} \subseteq D^{nn}$). Состояние ВП $\hat{\breve{D}}^{T}$, где имеют место только ненаблюдаемые данные, исключим из рассмотрения. Совокупную функциональность Д-операторов $\dot{O}$ и $\tilde{O}$ обозначим, как O, а входные и выходные данные – как D и D'. В результате получаем Д-оператор $(D)O(D')$, где $D = (\dot{D}'' \cup \tilde{D}^{\text{СБ}})$, $D' = (\tilde{D}''' \cup \dot{D}^{\text{СБ}})$.

Поскольку состав данных в состояниях ВП $\breve{D}^{T}$ и $\hat{D}^{T}$, не взирая на выполненные преобразования, в соответствии с леммой 1, не изменился, функциональность полученного Д-оператора по построению соответствует совокупной функциональности исходной композиции, а все ненаблюдаемые данные, в соответствии с определением 1, обрабатываются, то, полученный Д-оператор, в соответствии с определением 5, эквивалентен исходной композиции и принадлежит к более высокому (укрупненному), уровню описания алгоритма.

Если функциональность каждого из Д-операторов, входящих в композицию $(\dot{D})\dot{O}(\dot{D}') * (\tilde{D})\tilde{O}(\tilde{D}')$ и обрабатывающих, в соответствии с определением 1, данные $\dot{D}^{o}$ и $\tilde{D}^{o}$, задана и согласована с этими данными, то у производного Д-оператора, который по построению обрабатывает множество данных $D^{o} = \dot{D}^{o} \cup \tilde{D}^{o}$, и имеет функциональность, совпадающую с совокупной функциональностью исходных, функциональность тоже задана, а обрабатываемые данные с ней согласованы.

Теорема доказана.

Следствие 1. Из обобщения результата доказанного в теореме легко увидеть, что в результате объединения нескольких произвольных Д-операторов, организованных в произвольную последовательность, будет получен производный Д-оператор

$$(D_1)O_1(D_1') * \ldots * (D_k)O_k(D_k') =$$
$$= (D)O(D'),$$

функциональность которого соответствует совокупной функциональности исходных. Если в каждом из исходных Д-операторов функциональность задана и согласована с обрабатываемыми данными, то она задана и согласована с обрабатываемыми данными и в производном Д-операторе.

Следствие 2. Состав данных на входе и выходе производного Д-оператора однозначно определяется составом данных на входах и выходах исходных Д-операторов и в общем случае

$$D = (D_1'' \cup D_2^{\text{СБ}} \cup \ldots \cup D_k^{\text{СБ}}),$$

$$D' = (D_k''' \cup D_1'^{\text{СБ}} \cup \ldots \cup D_{k-1}'^{\text{СБ}}).$$

Следствие 3. Композиция предикатов (см. определение операции композиции) преобразуется в предикат

$$(D_1^{\pi})P_1(\alpha_1) \circ \ldots \circ (D_k^{\pi})P_k(\alpha_k) = (D^{\pi})P(\alpha),$$

где $\alpha_1 \circ \ldots \circ \alpha_k = \alpha$ ($\circ$ – логическая операция), $(D_1^{\pi} \cup \ldots \cup D_k^{\pi}) = D^{\pi}$.

Теперь покажем возможность построения производных Д-операторов в результате декомпозиции исходного. Опираясь на определение 1, будем полагать, что Д-оператор не элементарный, если его функциональность реализуется несколькими элементарными операциями.

Теорема 2. Любой не элементарный Д-оператор $(D)O(D')$ (за исключением $(D)O^{\text{д}}(D')$, $(D)O^{u}(D')$) может быть декомпозирован – представлен на более детальном уровне описания в виде композиции Д-операторов $(\dot{D})\dot{O}(\dot{D}') * (\tilde{D})\tilde{O}(\tilde{D}')$, которая будет ему эквивалентна, если на входах и выходах производных

Д-операторов специфицированы следующие данные:

$$\dot{D}=(D\setminus\tilde{D}^{\text{СБ}})\cup\dot{D}^{\text{Л}},\ \dot{D}'=\dot{D}^{\text{СБ}}\cup D^{\text{СВ}}\cup\dot{D}'^{\text{Л}},$$

$$\tilde{D}=\tilde{D}^{\text{СБ}}\cup D^{\text{СВ}}\cup\tilde{D}^{\text{Л}},\ \tilde{D}'=(D'\setminus\dot{D}^{\text{СБ}})\cup\tilde{D}'^{\text{Л}}.$$

Доказательство. Воспользовавшись тем, что, по условию теоремы, Д-оператор $(D)O(D')$ не элементарный, будем полагать, что его функциональность представляет собой два этапа, которые реализуются Д-операторами $\dot{O}$, $\tilde{O}$, и которые связаны операцией композиции, в соответствии с её определением. В результате такого построения получаем новое состояние ВП $\hat{\tilde{D}}^{T}$ (см. определение 3), в котором все данные ненаблюдаемые.

Переведем, с учетом определения 4, ненаблюдаемые данные

$$\dot{D}^{\text{Л}},\dot{D}'^{\text{Л}},\tilde{D}^{\text{Л}},\tilde{D}'^{\text{Л}},D^{\text{СВ}}\subseteq D^{nn}$$

в разряд наблюдаемых, а полученную композицию Д-операторов запишем в виде

$$(D,\dot{D}^{\text{Л}})\dot{O}(\dot{D}'^{\text{Л}},D^{\text{СВ}})*$$
$$*(D^{\text{СВ}},\tilde{D}^{\text{Л}})\tilde{O}(D',\tilde{D}'^{\text{Л}}),$$

где $(D,\dot{D}^{\text{Л}})\subseteq\hat{D}^{T}$, $(\dot{D}'^{\text{Л}},D^{\text{СВ}},\tilde{D}^{\text{Л}})\subseteq\hat{\tilde{D}}^{T}$, $(D',\tilde{D}'^{\text{Л}})\subseteq\breve{D}^{T}$.

В общем случае в результате разбиения функциональности на некоторое подмножество данных $\tilde{D}^{\text{СБ}}\subseteq D$ будет обрабатываться на втором этапе, а некоторое подмножество $\dot{D}^{\text{СБ}}\subseteq D'$ является результатом выполнения только первого этапа. Эти подмножества, в соответствии с определением 4, – собственные данные Д-операторов $\dot{O}$ и $\tilde{O}$, соответственно.

Исходя из этого, переместим спецификации этих данных таким образом, что будет получена композиция в виде

$$(\dot{D}'',\dot{D}^{\text{Л}})\dot{O}(\dot{D}^{\text{СБ}},\dot{D}'^{\text{Л}},D^{\text{СВ}})*$$
$$*(D^{\text{СВ}},\tilde{D}^{\text{СБ}},\tilde{D}^{\text{Л}})\tilde{O}(\tilde{D}''',\tilde{D}^{\text{Л}}),,$$

где $\dot{D}''=D\setminus\tilde{D}^{\text{СБ}}$ и $\tilde{D}'''=D'\setminus\dot{D}^{\text{СБ}}$.

Так как в полученной композиции, в соответствии с леммой 1, состояния ВП $\breve{D}^{T}$ и $\hat{D}^{T}$ не изменились, каждый производный Д-оператор обрабатывает свои собственные данные, а совокупная функциональность производных Д-операторов (по построению) совпадает с функциональностью исходного, то, в соответствии с определением 5, производная композиция и исходный Д-оператор эквивалентны, а полученная композиция принадлежит к более низкому (детальному) уровню описания алгоритма.

Теорема доказана.

Следствие 1. Из обобщения результата, доказанного в теореме, легко увидеть, что не элементарный Д-оператор $(D)O(D')$ может быть декомпозирован, то есть, представлен в виде произвольной последовательности Д-операторов

$$(D)O(D')=(D_1)O_1(D_1')*\ldots*(D_k)O_k(D_k'),$$

совокупная функциональность которой совпадает с функциональностью исходного Д-оператора.

Следствие 2. Состав входных и выходных данных производных Д-операторов, для которых выполняются соотношения

$$D\subseteq(D_1\cup\ldots\cup D_k),\ D'\subseteq(D_1'\cup\ldots\cup D_k')$$

зависит от способа декомпозиции и состава входных и выходных данных исходного Д-оператора.

Следствие 3. Предикаты декомпозируются в виде

$$(D^{\pi})P(\alpha)=(D_1^{\pi})P_1(\alpha_1)\circ\ldots\circ(D_k^{\pi})P_k(\alpha_k),$$

где $\alpha=\alpha_1\circ\ldots\circ\alpha_k$ ($\circ$ – логическая операция), $D^{\pi}=(D_1^{\pi}\cup\ldots\cup D_k^{\pi})$.

Доказанные теоремы являются обоснованием возможности реализации восходящей и нисходящей стратегий проектирования алгоритмов.

3. Полнота алгебраического аппарата

Прежде, чем перейти к доказательству полноты алгебры, уточним некоторые использованные выше понятия посредством следующих рассуждений.

Алгоритм решения некоторой задачи – это некоторый Д-оператор $(D)A(D')$, который обладает функциональностью (см. определение 1), обеспечивающей решение этой задачи. При нисходящем проектировании алгоритм декомпозируется до получения некоторых элементарных Д-операторов, а при восходящем строится, начиная с них.

Учитывая это, определим элементарные Д-операторы.

Определение 6. Элементарными называются Д-операторы $(D)O^{э}(D')$, не подлежащие декомпозиции, функциональность которых задана и согласована с обрабатываемыми данными. В случае проектирования от данных на входах и выходах элементарных Д-операторов специфицированы только элементарные данные $D^{э} \subseteq D$ и/или $D'^{э} \subseteq D'$. Элементарные Д-операторы образуют конечное множество $X^{Э} \subset X$, где X – множество Д-операторов.

Кроме того, разработка осуществляется, в соответствии со спецификой алгоритмов, с использованием двух подходов: от управления и от данных.

При первом подходе, ориентированном на разработку управляющих (и подобных) систем, последовательность подзадач, на которые разбивается алгоритм, зависит, обычно, от особенностей функционирования объекта управления.

При втором подходе, ориентированном на разработку информационных (и подобных) систем, на последовательность выполнения упомянутых подзадач существенным образом влияет структура иерархических данных. В этом случае проектирование осуществляется до достижения такого уровня представления алгоритма, на котором все иерархические данные детализованы до уровня элементарных.

Доказательство полноты алгебры предварим следующей леммой.

Лемма 2 (о функциональной полноте алгебры данных). Произвольные иерархические данные $D^{и} \subseteq S^{и}$ могут быть получены из наперед заданного конечного множества элементарных данных $S^{Э} = \{D_1^{э}, D_2^{э}, ..., D_p^{э}\}$ и детализованы до получения элементарных данных, если множество $S^{Э}$ включает элементарные данные последнего (k-го) уровня представления любых иерархических данных $D_i^{и} \subseteq S^{и}$.

Доказательство. Пусть задано некоторое конечное множество элементарных данных $S^{Э} = \{D_1^{э}, D_2^{э}, ..., D_p^{э}\}$ и произвольные иерархические данные $D^{и} \subseteq S^{и}$.

Выбрав из множества $S^{Э}$ подмножество $S''^{Э}$, элементы которого предположительно образуют нижний слой иерархического множества $D^{и}$, построим из них с помощью операции укрупнения $*\{D_i^{э}, ..., D_j^{э}\} = {}_{k-1}D_i$ все данные $(k-1)$-го уровня иерархии. В результате получаем множество данных $\{{}_{k-1}D_1, ..., {}_{k-1}D_s\}$. Далее, действуя аналогично, будем целенаправленно строить данные на всех вышележащих уровнях до получения $D^{и}$.

Если сделанные выше и на всех этапах построения иерархических данных предположения не реализуются, то есть полученные на $k-1$ (${}_{k-1}D_i$) или любом вышележащем уровне иерархии данные (в том числе $D^{и}$) не соответствуют требуемым, то возвращаемся на любой из уровней для его корректировки. В случае необходимости, выбираем другое множество элементарных данных $S'''^{Э}$. Поскольку, в соответствии с определением 2 и условием леммы, множество $S^{Э}$ конечно, а множество $S'^{Э} \subseteq S^{Э}$ существует, то за конечное число попыток множество $S'^{Э}$ будет найдено, а иерархические данные $D^{и}$ будут получены.

Применяя операцию детализации данных к данным $D^{и}$ $\overline{*} D = \{D_{1_1}, ..., D_{p_1}\}$ и на всех уровнях их иерархии, получаем, по условию теоремы и, в соответствии с определением 2, множество элементарных данных $\{D_1^{э}, D_2^{э}, ..., D_r^{э}\} = S'^{Э} \subseteq S^{Э}$.

Лемма доказана.

Покажем возможность проектирования произвольного алгоритма $(D)A(D')$ полагая, что он принадлежит некоторому множеству алгоритмов $\mathcal{A}$.

Теорема 3 (о функциональной полноте). В рамках САА\Д произвольный алгоритм $(D)A(D') \in \mathcal{A}$, может быть построен от управления, если заданное множество элементарных Д-операторов $X^{\text{Э}}$ достаточно для реализации функциональности любого алгоритма из $\mathcal{A}$, а алгебра логики функционально полна. Если, при этом, $D^{\text{И}} \subseteq D$ и/или $D'^{\text{И}} \subseteq D'$, алгебра данных функционально полна, и для любых элементарных данных $D_i^{\text{э}}, D_i'^{\text{э}} \in S^{\text{э}}$ в множестве $X^{\text{Э}}$ найдутся элементарные Д-операторы $(D)O^{\text{э}}(D')$ такие, что $D_i^{\text{э}} \subseteq D$ и/или $D_i'^{\text{э}} \subseteq D'$, то алгоритм может быть построен от данных. Процесс проектирования может осуществляться с использованием как нисходящей, так и восходящей стратегий, и на всех его этапах функциональность производных Д-операторов будет согласована с обрабатываемыми данными.

Доказательство. Зададимся конечным множеством элементарных Д-операторов

$$X^{\text{Э}} = \{(D_1)O_1^{\text{э}}(D_1'),...,(D_n)O_n^{\text{э}}(D_n')\}$$

и конечными множествами элементарных данных

$$S'^{\text{Э}} = \{D_1^{\text{э}},..., D_s^{\text{э}}\},\ S''^{\text{Э}} = \{D_1'^{\text{э}},..., D_p'^{\text{э}}\},$$

где $S'^{\text{Э}} \cup S''^{\text{Э}} = S^{\text{Э}}$, такими, что они соответствуют условиям теоремы.

Рассмотрим реализацию нисходящей стратегии проектирования алгоритма $(D)A(D') \in \mathcal{A}$.

Исходя из теоремы 2 и следствия 1 из неё, запишем алгоритм в виде КС

$$(D)A(D') =$$
$$=({}_1D_1)_1O_1({}_1D_1') * ... *({}_1D_k)_1O_k({}_1D_k'),$$

где левый нижний индекс указывает шаг проектирования. Производные Д-операторы выбираются исходя из предположения, что совокупная их функциональность обеспечивает требуемую алгоритмом функциональность.

В случае реализации подхода от данных, когда, в соответствии с условием теоремы, для алгоритма $(D)A(D')$ выполняются соотношения $D^{\text{И}} \subseteq D$ и/или $D'^{\text{И}} \subseteq D'$. Процесс проектирования начинается с детализации иерархических данных $\bar{*} D^{\text{И}} = \{D_1,..., D_p\}$, $\bar{*} D'^{\text{И}} = \{D_1',..., D_r'\}$, а декомпозиция осуществляется аналогично вышерассмотренному случаю, с тем дополнением, что, с учетом следствия 2 из теоремы 2, для любых $D_i \in D^{\text{И}}$ и $D_i' \in D'^{\text{И}}$ выполняются соотношения $D_i \subseteq_1 D_p$ и $D_i' \subseteq_1 D_s'$.

Если в результате декомпозиции получены Д-операторы вида $({}_1D_i)_1O_i^{\text{Д}}({}_1D_i')$ или $({}_1D_i)_1O_i^{\text{И}}({}_1D_i')$, то, в соответствии с (1) и (2), они заменяются соответствующими операциями алгебры, в результате чего получаем РС.

Применяя описанный выше подход ко всем производным Д-операторам, получаем более детальное описание каждого из них в виде

$$({}_1D_i)_1O_i({}_1D_i') =({}_2D_1)_2O_1({}_2D_1') *$$
$$* ... *({}_2D_s)_2O_s({}_2D_s').$$

Если на предыдущем шаге декомпозиции была получена РС, то декомпозируются Д-операторы и предикаты, входившие в операции алгебры. Последние, в соответствии со следствием 3 из теоремы 2, записываются в виде КС

$$({}_1D_i^{\pi})P(\alpha) =({}_2D_1^{\pi})_2P_1(\alpha) \circ ... \circ({}_2D_p^{\pi})_2P_p(\alpha).$$

Процесс проектирования от управления продолжается до получения совокупности СА, каждая из которых записывается в виде

$$({}_{k-1}D_i)_{k-1}O_i({}_{k-1}D_i') =({}_kD_1)_kO_1^{\text{э}}({}_kD_1') *$$
$$* ... *({}_kD_p)_kO_p^{\text{э}}({}_kD_p'),$$

где все производные Д-операторы элементарные.

Поскольку по условию теоремы алгебра данных полна, то, в соответствии с леммой 2, в результате детализации иерархических данных будет получено множество элементарных данных $S'''^{э} \subseteq S^{э}$. Так как, по условию теоремы для любых элементарных данных $D_i^{э}, D_i'^{э} \in S'''^{э}$ выполняются соотношения $D_i^{э} \subseteq_k D_j$ и/или $D_i'^{э} \subseteq_k D_s'$, то процесс проектирования от данных завершится после получения аналогичной совокупности СА, в которой производные Д-операторы обрабатывают все элементарные данные из $S'''^{э}$.

Перейдем к рассмотрению восходящей стратегии проектирования.

В соответствии со следствием 1 из теоремы 1, построим совокупность СА, каждая из которых записывается в виде

$$({}_kD_1)_k O_1^{э}({}_kD_1') * \ldots * ({}_kD_p)_k O_p^{э}({}_kD_p') = \\ = ({}_{k-1}D_i)_{k-1} O_i ({}_{k-1}D_i'),$$

где элементарные Д-операторы выбираются и располагаются в некоторой последовательности исходя из предположения, что их совокупная функциональность соответствует требуемой функциональности производного Д-оператора $({}_{k-1}D_i)_{k-1} O_i ({}_{k-1}D_i')$.

Процесс проектирования от данных начинается с построения иерархических данных, что для каждого случая записывается с использованием операции укрупнения данных в виде $*\{D_i^{э},..., D_j^{э}\} = D$, $*\{D_r'^{э},..., D_s'^{э}\} = D'$, исходя из предположения, что производные данные D и D' соответствуют $(k-1)$-му уровню иерархии данных. Построение совокупности СА выполняется аналогично вышерассмотренному случаю с тем дополнением, что, с учетом следствия 2 из теоремы 1 элементарные Д-операторы выбираются таким образом, что бы для производных иерархических данных D и D' выполняются соотношения $D \subseteq_{k-1} D_i$, $D' \subseteq_{k-1} D_i'$.

Если требуется реализовать операцию алгебры, то эта операция записывается с использованием элементарных Д-операторов и предиката. Последний (в случае необходимости), предварительно записывается, в соответствии со следствием 3 из теоремы 1, в виде КС

$$({}_kD_1^{э})_k P_1(\alpha) \circ \ldots \circ ({}_kD_s^{э})_k P_s(\alpha) = \\ = ({}_{k-1}D_i^{\pi}) P(\alpha).$$

Полученная таким образом операция алгебры записывается, в соответствии с (1) или (2), в виде соответствующего Д-оператора.

Применяя описанный выше подход ко всем производным Д-операторам, переходим на более высокий уровень описания алгоритма, на котором каждый из Д-операторов записывается в виде

$$({}_{k-1}D_1)_{k-1} O_1 ({}_{k-1}D_1') * \ldots * ({}_{k-1}D_s)_{k-1} O_s ({}_{k-1}D_s') = \\ = ({}_{k-2}D_i)_{k-2} O_i ({}_{k-2}D_i').$$

Процесс проектирования продолжается с использованием производных Д-операторов аналогично и приводит к получению все более укрупненного описания алгоритма. В случае проектирования от управления, этот процесс завершится построением СА

$$({}_1D_1)_1 O_1 ({}_1D_1') * \ldots * ({}_1D_k)_1 O_k ({}_1D_k') = \\ = (D) A (D').$$

Поскольку по условию теоремы алгебра данных полна, то, в соответствии с леммой 2, в результате укрупнения иерархических данных будут получены $D^{и}$ и $D'^{и}$, а процесс проектирования от данных завершится построением Д-оператора $(D)A(D')$ такого, что $D^{и} \subseteq D$ и $D'^{и} \subseteq D'$.

Если во всех рассмотренных случаях осуществить свертку СА (последовательно подставить элементарные Д-операторы во все схемы снизу вверх), то, при условии выполнения предположений, которые делались в процессе проектирования, будет получена последовательность элементарных Д-операторов, совокупная функциональность которой по построению обеспечивает решение требуемой задачи, а при разработке от данных – решение задачи с учетом структуры иерархических данных.

Поскольку достаточно произвольно осуществляется декомпозиция и объединение Д-операторов, выбор элементарных Д-операторов и данных, то во всех рассмотренных случаях возможна ситуация, когда сделанные предположения не выполнятся. В такой ситуации осуществляется корректировка полученных результатов, начиная с любого шага проектирования.

Так как по условиям теоремы множество $X^{Э}$ конечно и достаточно для реализации функциональности произвольного алгоритма, а для всех элементарных данных $D_i^{Э}, D_i'^{Э} \in S^{Э}$ в множестве $X^{Э}$ найдутся Д-операторы $(D)O^{Э}(D')$ такие, что $D_i^{Э} \subseteq D$ и/или $D_i'^{Э} \subseteq D'$, то во всех рассмотренных случаях в результате конечного числа корректировок искомый результат будет получен. При этом, поскольку по условию теоремы алгебра логики полна (доказательство полноты приведено, например, в работе [6]), то в процессе проектирования реализуемы любые логические условия.

Таким образом, показана реализуемость разработки произвольного алгоритма $(D)A(D')$ от управления и от данных в рамках как нисходящей, так и восходящей стратегий проектирования.

При этом, в соответствии с определением 6, функциональность элементарных Д-операторов задана и согласована с обрабатываемыми данными. В соответствии с этим и следствием 1 из теоремы 1, при восходящем проектировании функциональность каждого Д-оператора $({}_{k-1}D_i)_{k-1}O_i({}_{k-1}D_i')$ будет задана и согласована с обрабатываемыми данными. Так как это свойство сохранится на всех вышележащих уровнях проектирования, то функциональность получаемых на каждом из них Д-операторов (включая $(D)A(D')$), будет согласована с обрабатываемыми данными.

Легко увидеть, что применение следствия 1 из теоремы 1 к СА, полученным в случае нисходящего проектирования, на нижнем и всех последующих уровнях описания алгоритма приведет к такому же результату.

Теорема доказана.

Выводы

Показана возможность проектирования в рамках алгебры алгоритмов с данными произвольного алгоритма $(D)A(D')$ от управления и от данных, как с использованием нисходящей, так и восходящей стратегий. При этом предложенная алгебра обеспечивает в проектируемом алгоритме согласованность потоков управления и обрабатываемых данных на всех этапах разработки. Из доказанной теоремы 3 легко увидеть, что рассмотренные стратегии и подходы к проектированию могут использоваться совместно. Возможность автоматизированного проектирования алгоритмов средствами интегрированного инструментария проектирования и синтеза программ продемонстрирована в работе [9].

1. *Вирт Н.* Алгоритмы + структуры данных = программы : перев. с англ. – М.: Мир, 1985. – 656 с.
2. *Турский В.* Методология программирования: пер. с англ. – М.: Мир, 1981. – 264 с.
3. *Замулин А.В.* Формальные методы спецификации программ. – Новосибирск: Новосибирский государственный университет, 1999. – 97 с.
4. *Акуловский В.Г.* Алгебра алгоритмов, базирующаяся на данных // Кибернетика и системный анализ. – 2012. – № 2. – С. 151–166.
5. *Глушков В.М., Цейтлин Г. Е., Ющенко Е.Л.* Алгебра. Языки. Программирование. 3-е изд., перераб. и доп. – К.: Наукова думка, 1989. – 376 с.
6. *Цейтлин Г.Е.* Введение в алгоритмику. – К.: Сфера, 1998. – 310 с.
7. *Дорошенко А.Ю., Акуловський В.Г.* Висхідне проектування алгоритмів при алгеброалгоритмічному підході // Вісник Київського національного університету імені Тараса Шевченка. Серія фізико-математичні науки. – 2012. – Вип. 1. – С. 167–172.
8. *Дорошенко А.Е., Акуловский В.Г.* Нисходящее проектирование алгоритмов в рамках алгеброалгоритмического подхода // Математичні машини і системи. – 2012. – № 3. – С. 97–102.

9. *Акуловский В.Г., Дорошенко А.Е., Яценко Е.А.* Реализация средств проектирования и генерации программ на основе алгебры алгоритмов с данными // Проблеми програмування. – 2015. – № 2. – С. 41–51.

References

1. Wirth, N. (1985) *Algorithms + Data Structures = Programs*. Moscow: Mir. (in Russian)
2. Turski, W. (1981) *Methodology of programming*. Moscow: Mir. (in Russian)
3. Zamulin, A.V. (1999) *Formal methods of program specification*. Novosibirsk: Novosibirsk State University. (in Russian)
4. Akulovskiy, V.G. (2012) Data based algorithmic algebra. *Cybernetics and systems analysis*. (2). P. 151–166. (in Russian)
5. Glushkov, V.M., Tseitlin, G.E. & Yushchenko, E.L. (1989) *Algebra. Languages. Programming. 3rd edition.* Kiev: Naukova dumka. (in Russian)
6. Tseitlin, G.E. (1998) *Introduction to algorithmics*. Kiev: Sfera. (in Russian)
7. Doroshenko, A.Yu. & Akulovskiy, V.G. (2012) The bottom-up design of algorithms at algebra-algorithmic approach. *Bulletin of Taras Shevchenko National University of Kyiv. Series Physics & Mathematics*. (1). P. 167–172. (in Ukrainian)
8. Doroshenko, A.Yu. & Akulovskiy, V.G. (2012) The top-down design of algorithms at algebra-algorithmic approach. *Mathematical machines and systems*. (3). P. 97–102. (in Russian)
9. Akulovskiy, V.G., Doroshenko, A.Yu. & Yatsenko, O.A. (2015) Implementation of tools for designing and generating of programs on the basis of algebra of algorithms with data. *Problems in programming*. (2). P. 41–51. (in Russian)

Получено 21.09.2016

***Об авторах*:**

Акуловский Валерий Григорьевич
кандидат технических наук, доцент кафедры информационных систем и технологий Академии таможенной службы Украины.
Количество научных публикаций в украинских изданиях – более 30.
Количество научных публикаций в зарубежных изданиях – 4.
Индекс Хирша – 4.
http://orcid.org/0000-0001-6180-0763,

Дорошенко Анатолий Ефимович,
доктор физико-математических наук, профессор, заведующий отделом теории компьютерных вычислений Института программных систем НАН Украины, профессор кафедры автоматики и управления в технических системах НТУУ “КПИ”.
Количество научных публикаций в украинских изданиях – более 150.
Количество научных публикаций в зарубежных изданиях – более 30.
Индекс Хирша – 3.
http://orcid.org /0000-0002-8435-1451,

Яценко Елена Анатольевна,
кандидат физико-математических наук, старший научный сотрудник Института программных систем НАН Украины.
Количество научных публикаций в украинских изданиях – 30.
Количество научных публикаций в иностранных изданиях – 14.
http://orcid.org/0000-0002-4700-6704.

Место работы авторов:

Академия таможенной службы Украины.
49000, Днепропетровск,
ул. Дзержинского, 2/4.
E-mail: valeryakulovskiy@rambler.ru

Институт программных систем
НАН Украины. 03187, Киев,
проспект Академика Глушкова, 40.
Тел.: (044) 526 3559.
E-mail: doroshenkoanatoliy2@gmail.com,
oayat@ukr.net