УДК 004.42

СОГЛАСОВАННОЕ ОПИСАНИЕ АЛГОРИТМОВ В РАМКАХ АЛГЕБРАИЧЕСКОГО АППАРАТА

В.Г. Акуловский, А.Е. Дорошенко

Академия таможенной службы Украины.
49000, Днепропетровск, ул. Дзержинского 2/4.
Тел/Факс: (0562) 745 5596, (0562) 47 1791
E-mail: academy@amsu.dp.ua.
Інститут програмних систем НАН України,
03187, Київ, проспект Академіка Глушкова 40.
Тел.: 526 3559, e-mail: doroshenkoanatoliy2@gmail.com

Продемонстрирована (в самом общем случае) возможность согласованного описания потоков управления, данных и информационных связей в процессе разработки алгоритмов средствами трехосновной алгебраической системы (алгебры алгоритмов с данными). Показаны свойства получаемых схем алгоритмов.

The possibility of the concerted description of management streams is shown in general, data and informative connections in the process of algorithms' development facilities of the tribasic algebraic system (algebras of algorithms with data) is shown (in most general case). The properties of the got charts of algorithms are rotined.

Введение

Поскольку алгоритм является многоаспектной сущностью, то актуальна задача такого его описания, в рамках которого отображаются основные его аспекты: потоки управления, структуры обрабатываемых данных и информационные связи. При этом на каждом этапе разработки эти аспекты должны быть согласованы между собой.

Учитывая важность роли данных в процессе разработки алгоритмов [1, 2], для решения этой задачи предложен алгебраический аппарат [3–5] в который, в результате модификации известной модели ЭВМ Глушкова [6], "встроены" данные.

Далее приведены некоторые элементы этого алгебраического аппарата необходимые решения указанной задачи.

Алгебра алгоритмов с данными

Упомянутый алгебраический аппарат представляет собой трехосновную алгебраическую систему $\text{САА}\backslash\text{Д}::=\langle\{U,L,D\};\Omega\rangle$ (алгебру алгоритмов с данными), где U – множество Д-операторов, L – множество логических условий, D – множество данных, Ω – базовый набор операций алгебры, включающий операции Ω_1, определенные на множестве U, Ω_2, определенные на множестве L, Ω_3, определенные на множестве D.

В рамках алгебры данные определены следующим образом.

Определение 1. Составными данными будем называть пару $D=\langle N_{A_j}, З\rangle$, где $З=\langle з_1,...,з_n\rangle$ – кортеж значений, простыми – $d=\langle N_{A_j}, з\rangle$, где з – значение, носитель которых N_{A_j}, однозначно идентифицируется адресом $A_j \in I$. В каждый момент времени данные хранят некоторый кортеж значений З, значение – з, определяющий текущее состояние данных.

Используемые далее типы данных соответствуют традиционному для языков программирования понятию и будут обозначаться f_i, где $f_i \in F = \{f_1, f_2,..., f_k\}$, а F – множество возможных типов данных.

Первичными структурами данных являются: составные данные (первичные записи) D^S такие, что $З=\langle з_1^{f_x},...,з_k^{f_x}\rangle$ где f_x – произвольный тип данных, и одномерные массивы (векторы) ${}^{n}D^{f_i}$ такие, что $З=\langle з_1^{f_i},...,з_n^{f_i}\rangle$, где f_i – конкретный тип данных, n – число этих значений (размерность массива). Заметим, что вектор может интерпретироваться различным образом, например, как строка.

На множестве упорядоченных первичных структур данных, которые в общем случае обозначим D^S, определим операцию **объединения** (композиции) данных – "*". В результате применения этой операции к первичным структурам данных $D^S * D'^S = {}_1D_i^{З\Pi}$, в общем случае $D_1^S * ... * D_p^S = {}_1D_j^{З\Pi}$, получаем новую структуру данных, называемую записью, где левый нижний индекс – указывает уровень вложенности структур.

Применяя операцию композиции к полученным записям ${}_1\mathrm{D}_1^{ЗП} * ... * {}_1\mathrm{D}_s^{ЗП} = {}_2\mathrm{D}_j^{ЗП}$, получаем новую запись второго уровня вложенности, в которую могут быть вложены как первичные, так и производные структуры данных. Продолжая такой процесс можем получать записи ${}_n\mathrm{D}_j^{ЗП}$ с практически неограниченной вложенностью. Получаемые таким образом записи на каждом уровне вложенности могут трактоваться как кортеж семейств множеств ${}_n\mathrm{D}_j^{ЗП} = \left\langle {}_{n-1}\mathrm{D}_1^{ЗП},...,{}_{n-1}\mathrm{D}_k^{ЗП} \right\rangle$ таких, что для любого ${}_{n-2}\mathrm{D}_{i_j}^{ЗП} \in {}_{n-1}\mathrm{D}_i^{ЗП}$ выполняется ${}_{n-2}\mathrm{D}_{i_j}^{ЗП} \notin {}_{n-1}\mathrm{D}_p^{ЗП}$ для всех $p \neq i$.

Для построения производных структур данных введем производные операции, первой из которых является операция **итерации данных**, полученная в результате обобщения операции композиции на n одинаковых элементов и записываемая в виде "$\{\}^n$".

Применяя эту операцию к некоторому одномерному массиву размера n_1 получаем двумерный массив $\{{}^{n_1}D^{f_i}\}^{n_2} = {}^{n_2,n_1}\mathrm{D}^{f_i}$, к двумерному – трехмерный массив ${}^{n_3,n_2,n_1}\mathrm{D}^{f_i}$ и т. д., на k-ом шаге получаем $\{{}^{n_{k-1},...n_2,n_1}D^{f_i}\}^{n_k} = {}^{n_k,...n_2,n_1}\mathrm{D}^{f_i}$ k-мерный массив.

Распространив операцию итерации данных на множество записей, получаем $\{D^{ЗП}\}^n = {}^n D^{ЗП}$ одномерный массив записей (в дальнейшем массив записей). Применяя эту операцию к массиву записей k раз, получаем многомерный массив записей в виде $\{{}^{n_{k-1},...n_1}D^{ЗП}\}^{n_k} = {}^{n_k,...n_1}\mathrm{D}^{ЗП}$.

Введенная операция позволила получить производные структуры данных многомерные массивы, одномерные и многомерные массивы записей. Операция композиции распространенная на полученные производные структуры данных, позволяет организовывать записи, включающие в свой состав простые данные, одномерные и многомерные массивы, одномерные и многомерные массивы записей, то есть получать структуры данных произвольной сложности вложенные на произвольную глубину.

Для случая, когда необходимо описывать в виде единого целого произвольно расположенные, в частности, семантически несвязанные данные, введем операцию **группировки данных**, которую обозначим "$\dot{\cup}$". В результате применения этой операции к произвольным структурам данных $D'^S \dot{\cup} D''^S = {}_1D_i^\Gamma$, а в общем случае $D_1^S \dot{\cup} ... \dot{\cup} D_p^S = {}_1D_i^\Gamma$, получаем новые данные, называющиеся первичными группами данных. Распространив эту операцию на множество групп ${}_1D_1^\Gamma \dot{\cup} ... \dot{\cup} {}_1D_m^\Gamma = {}_2D_1^\Gamma$ получаем новую группу данных второго уровня вложенности, в которую могут быть вложены как первичные, так и производные группы.

Элементы группы данных могут быть, как упорядочены, так и не упорядочены, а отличие их от структур данных состоит в том, что на множестве групп данных операция композиции не определена. Последовательное применение операции группировки к группам данных ${}_{n-1}\mathrm{D}_1^\Gamma \dot{\cup} ... \dot{\cup} {}_{n-1}\mathrm{D}_k^\Gamma = {}_n\mathrm{D}_j^\Gamma$ позволяет получить вложенность групп данных на практически произвольную глубину.

На любом, начиная со второго уровня вложенности групп данных, они могут трактоваться как семейство множеств ${}_n\mathrm{D}_j^\Gamma = \{{}_{n-1}\mathrm{D}_1^\Gamma,...,{}_{n-1}\mathrm{D}_k^\Gamma\}$ таких, что для любого ${}_{n-2}\mathrm{D}_{i_j}^\Gamma \in {}_{n-1}\mathrm{D}_i^\Gamma$ допустимо как ${}_{n-2}\mathrm{D}_{i_j}^\Gamma \notin {}_{n-1}\mathrm{D}_p^\Gamma$, так и ${}_{n-2}\mathrm{D}_{i_j}^\Gamma \in {}_{n-1}\mathrm{D}_p^\Gamma$ для всех $p \neq i$.

Для детализации данных введем операции.

Операция **разгруппировки** "$\overline{\dot{\cup}}$", применение которой к группе данных

$$\overline{\dot{\cup}}_n D_i^\Gamma = {}_{n-1}D_1^\Gamma,...,{}_{n-1}D_p^\Gamma \tag{1}$$

порождает группы данных предшествующего уровня вложенности. В частных случаях и в случае первичной группы данных получаем структуры данных и\или простые данные.

Вторая такая операция – это операция **детализации (декомпозиции)** данных, обозначенная – "$\overline{*}$".

Применение этой операции к полученным выше структурам данных выполняется следующим образом:

– из многомерного массива записей получаем n_k массивов записей

$$\overline{*}({}^{n_k,...n}D^{ЗП}) = {}^{n_{k-1},...n_1,n}D_1^{ЗП},...,{}^{n_{k-1},...n_1,n}D_{n_k}^{ЗП}; \tag{2}$$

– из одномерного массива записей получаем n записей

$$\overline{*}({}^nD^{ЗП}) = D_1^{ЗП},...,D_n^{ЗП}; \tag{3}$$

– из записей получаем записи

$$\overline{*}_n D_i^{ЗП} = {}_{n-1}D_1^{ЗП},...,{}_{n-1}D_m^{ЗП} \tag{4}$$

или/и любые другие структуры данных (массивы записей, массивы, простые данные). В случае первичной записи $\bar{*}\mathrm{D}_1^s = d_1^{f_x},\ldots d_n^{f_x}$ – кортеж простых данных произвольного типа;

– из многомерного массива получаем $\mathrm{n_k}$ массивов

$$\bar{*}(^{\mathrm{n_k},\ldots\mathrm{n_1}}\mathrm{D}^{f_i}) = {}^{\mathrm{n_{k-1}},\ldots\mathrm{n_1}}\mathrm{D}_1^{f_i},\ldots,{}^{\mathrm{n_{k-1}},\ldots\mathrm{n_1}}\mathrm{D}_{\mathrm{n_k}}^{f_i}; \tag{5}$$

– из одномерного массива получаем кортеж из n простых данных типа $\mathrm{f_i}$

$$\bar{*}(^{\mathrm{n}}\mathrm{D}^{f_i}) = d_1^{f_i},\ldots,d_n^{f_i}. \tag{6}$$

Приведенные операции позволяют детализовать все возможные структуры данных. В результате такой декомпозиции массивы разбиваются на образующие их компоненты. Однако необходимость осуществлять детализацию сложных глубоко вложенных структур данных требует дополнительных средств. Достаточно легко увидеть, что на основе введенных легко могут строиться производные операции.

В частности, для групп данных операция

$${}_r\bar{*}_s({}_nD_i^{\Gamma}) = {}_{n-1}D_1^{\Gamma},\ldots,{}_{n-1}D_r^{\Gamma},{}_nD_i'^{\Gamma},{}_{n-1}D_s^{\Gamma},\ldots,{}_{n-1}D_p^{\Gamma}, \tag{7}$$

где ${}_nD_i'^{\Gamma} = {}_{n-1}D_{r+1}^{\Gamma} \dot{\cup} \ldots \dot{\cup} {}_{n-1}D_{s-1}^{\Gamma}$, отделяет r первых и s последних групп данных предыдущего уровня их вложенности. Ограничением на применение этой операции является соотношение $\mathrm{r} \le \mathrm{s}$. Заметим, что при $\mathrm{r} = \mathrm{s}$ операции вырождаются в исходную операцию разгруппировки (см. (1)). Заметим так же, что эти операции, очевидно, могут использоваться и в случае, когда детализуемые группы данных включают не только группы, а и любые структуры данных.

Для записей производными операциями детализации данных являются:

$$\bar{*}_{p_1,p_2,\ldots p_s}({}_nD_i^{3\Pi}) = {}_nD_{i_1}^{3\Pi},\ldots,{}_nD_{i_s}^{3\Pi}, \tag{8}$$

где ${}_nD_{i_j}^{3\Pi} = {}_{n-1}D_{p_1+\ldots+p_{j-1}+1}^{3\Pi} \dot{\cup} \ldots \dot{\cup} {}_{n-1}D_{p_1+\ldots+p_j}^{3\Pi}$;

$${}_r\bar{*}_s({}_nD_i^{3\Pi}) = {}_{n-1}D_1^{3\Pi},\ldots,{}_{n-1}D_r^{3\Pi},{}_nD_i'^{3\Pi},{}_{n-1}D_s^{3\Pi},\ldots,{}_{n-1}D_p^{3\Pi}, \tag{9}$$

где ${}_nD_i'^{3\Pi} = {}_{n-1}D_{r+1}^{3\Pi} * \ldots * {}_{n-1}D_{s-1}^{3\Pi}$.

Для массивов и массивов записей, которые в общем многомерном случае обозначим $^{\mathrm{n_k},\ldots,\mathrm{n_1}}\mathrm{D}$, производными операциями детализации данных являются:

$$\bar{*}_{p_1,p_2,\ldots p_s}({}^{s_k,\ldots,s_1}D_j^{f_i}) = {}^{p_1,s_{k-1},\ldots,s_1}D_{j_1}^{f_i},\ldots,{}^{p_s,s_{k-1},\ldots,s_1}D_{j_s}^{f_i} \text{ и } \bar{*}_{p_1,p_2,\ldots p_s}({}^{s_k,\ldots,s_1}{}_nD_i^{3\Pi}) = {}^{p_1,s_{k-1},\ldots,s_1}{}_nD_{i_1}^{3\Pi},\ldots,{}^{p_s,s_{k-1},\ldots,s_1}{}_nD_{i_s}^{3\Pi}, \tag{10}$$

в результате выполнения которых каждый из исходных массивов распадается на s массивов таких, что $p_1 + p_2 + \ldots + p_s = s_k$, что является ограничением на применение этой операции.

Эта операция, примененная к одномерным массивам

$$\bar{*}_{p_1,p_2,\ldots p_s}({}^{s}D_j^{f_i}) = {}^{p_1}D_{j_1}^{f_i},\ldots,{}^{p_s}D_{j_s}^{f_i} \quad \text{и} \quad \bar{*}_{p_1,p_2,\ldots p_s}({}^{s}{}_nD_i^{3\Pi}) = {}^{p_1}{}_nD_{i_1}^{3\Pi},\ldots,{}^{p_s}{}_nD_{i_s}^{3\Pi}, \tag{11}$$

порождает s одномерных массивов, при условии, что $p_1 + p_2 + \ldots + p_s = s$.

Распространив производную операцию (4) на множество массивов, получаем следующий результат:

$${}_r\bar{*}_s({}^{s_k,\ldots,s_1}D_j^{f_i}) = {}^{s_{k-1},\ldots,s_1}D_{j_1}^{f_i},\ldots,{}^{s_{k-1},\ldots,s_1}D_{j_r}^{f_i},{}^{(s_k-r-s),\ldots,s_1}D_j'^{f_i},{}^{s_{k-1},\ldots,s_1}D_{j_{r+1}}^{f_i},\ldots,{}^{s_{k-1},\ldots,s_1}D_{j_{r+s}}^{f_i}$$

и (12)

$${}_r\bar{*}_s({}^{s_k,\ldots,s_1}{}_nD_i^{3\Pi}) = {}^{s_{k-1},\ldots,s_1}{}_{n-1}D_{i_1}^{3\Pi},\ldots,{}^{s_{k-1},\ldots,s_1}{}_{n-1}D_{i_r}^{3\Pi}\,{}^{(s_k-r-s),\ldots,s_1}{}_nD_i'^{3\Pi},{}^{s_{k-1},\ldots,s_1}{}_{n-1}D_{i_{r+1}}^{3\Pi},\ldots,{}^{s_{k-1},\ldots,s_1}{}_{n-1}D_{i_{r+s}}^{3\Pi},$$

при условии $r + s \le s_k$.

Введенные операции (1)–(12) позволяют разгруппировать (перегруппировать) группы и детализовать структуры данных произвольной сложности и размера вложенные на произвольную. Производные операции, набор которых, очевидно может быть расширен, упрощают процесс детализации сложно организованных данных.

К логическим операциям, определенным на множестве логических условий L, отнесены известные операции **дизъюнкция**, **конъюнкция отрицание** со своими таблицами истинности [6].

Фундаментальным элементом описания алгоритмов, в рамках этого аппарата, являются Д-операторы $(D)X(D')$, обрабатывающие данные и определенные в работе [4] следующим образом.

Определение 2. На входе Д-оператора $(D)X(D')$ специфицировано множество входных данных D (в частном случае простые данные d), обрабатывая которые он в общем случае изменяет множество выходных (специфицированных на его выходе) данных D' (в частном случае простых данных d'). Для множеств входных и выходных данных таких, что $D \neq \varnothing$ и $D' \neq \varnothing$, может выполняться как соотношение $D \cap D' \neq \varnothing$ (в частности, $D \subseteq D'$ и $D' \subseteq D$), так и $D \cap D' = \varnothing$. В первом случае, изменяются значения подмножества входных данных $D \supseteq D'' = (D \cap D')$ (в частности, все входные данные, если $D \subseteq D'$). Во втором – значения входных данных остаются неизменными.

Определяющей для дальнейших рассуждений является операция **композиции Д-операторов** определенная следующим образом.

Композиция Д-операторов (операция обозначается "*") $(D)X(D')*(\tilde{D})\tilde{X}(\tilde{D}')$ означает последовательное выполнение сначала Д-оператора $(D)X(D')$ затем Д-оператора $(\tilde{D})\tilde{X}(\tilde{D}')$. То есть, Д-оператор $(D)X(D')$ непосредственно предшествует Д-оператору $(\tilde{D})\tilde{X}(\tilde{D}')$, а Д-оператор $(\tilde{D})\tilde{X}(\tilde{D}')$ непосредственно следует за Д-оператором $(D)X(D')$.

Естественное обобщение этой операции записывается в виде $(D_1)X_1(D_1')*...*(D_i)X_i(D_i')*...*(D_s)X_s(D_s')$.

Кроме операции композиции сигнатура алгебры Д-операторов включает множество других операций (см. [3, 4]), которые в данной работе только упоминаются.

На основе изложенного перейдем к решению поставленной задачи.

Разработка алгоритмов

Перед тем, как разрабатывать алгоритм, определим его.

Определение 3. Алгоритмом называется Д-оператор $(D)A(D')$, на входе которого специфицированы все глобальные данные подлежащие обработке, а на выходе все глобальные данные, являющиеся результатом этой обработки (результирующие).

Очевидно, что обрабатываемые – это определенные (инициализированные), то есть имеющие некоторые начальные значения данные, в частности константы, или данные, вводимые с внешних устройств (клавиатуры, магнитных дисков, аналого-цифровых преобразователей и т. д. и т. п.). Результирующие – это данные, являющиеся результатом работы программной системы, отображаемые (например, на экране дисплея) и/или сохраняемые на внешних устройствах.

Исходя из определения и приведенных соображений, алгоритм запишем в виде

$$(D^{\text{ИСХ}}, D^{\text{ИСХ\_}R})A(D^{\text{РЕЗ}}),$$

где $D^{\text{ИСХ}}$ – исходные, $D^{\text{ИСХ\_}R}$ – вводимые исходные данные, $D^{\text{РЕЗ}}$ – результирующие данные. При этом $\text{D}^{\text{ИСХ}}$ или $D^{\text{РЕЗ}}$ могут отсутствовать, а множество $D^{\text{РЕЗ}} \neq \varnothing$.

Рассмотрение процесс разработки алгоритма начнем со следующего требования предъявляемого к этому процессу. Наряду с реализацией известных принципов (формальности, абстрактности, иерархической упорядоченности и т. д.), в процессе разработки необходимо обеспечить, что бы все специфицированные в алгоритме данные были обработаны, а все обрабатываемые – специфицированы.

В качестве основы для организации процесса разработки воспользуемся теоремой, доказанной в работе [7], смысл которой, если отбросить не существенные для рассматриваемого случая подробности, состоит в следующем.

Произвольный Д-оператор $(D)X(D')$ может быть декомпозирован, то есть, представлен в виде эквивалентной ему композиции двух других Д-операторов $((D)X(D') = (\dot{D})\dot{X}(\dot{D}')*(\ddot{D})\ddot{X}(\ddot{D}')$. Обобщение этого результата позволяет ввести следующую форму представления Д-оператора, $(D)X(D') = (D_1)X_1(D_1')*(D_2)X_2(D_2')*...*(D_k)X_k(D_k')$, которая называется его композиционной схемой (КС), в которой $(D)X(D')$ – исходный, а $(D_1)X_1(D_1')$, $(D_2)X_2(D_2')$, …, $(D_k)X_k(D_k')$ – производные Д-операторы.

С другой стороны представление любого Д-оператора через образующие элементы системы $\langle\{U, L\};\{\Omega_1, \Omega_2\}\rangle$ назовем регулярной схемой этого Д-оператора (РСД). При этом в работе [4] показано, что любую операцию из Ω_1 на некотором этапе разработки можно трактовать как Д-оператор, а Д-оператор с соответствующей функциональностью, как операция алгебры Д-операторов. Из этого следует, что возможен переход от композиционных к регулярным схемам Д-операторов в процессе описания алгоритма.

Представление любых данных через образующие элементы системы $\langle D;\Omega_3\rangle$ посредством операций алгебры данных, назовем схемой данных (СД).

Представление любого Д-оператора через образующие элементы системы $\langle\{\text{U, L, D}\};\Omega\rangle$ назовем полной схемой этого Д-оператора (ПС).

Процесс разработки алгоритма начинается с этапа проектирования архитектуры программной системы, который играет ключевую роль, так как от качества его реализации в существенной мере зависят все последующие этапы и, в конечном счете, качество программного продукта. На этом этапе решаются две взаимосвязанные основные задачи:

1) определяется состав подсистем, образующих алгоритм, и специфицируются обрабатываемые и продуцируемые этими подсистемами данные;

2) специфицируются взаимосвязи между этими подсистемами.

Учитывая эти задачи подсистему определим следующим образом.

Определение 4. Подсистемой назовем Д-оператор $(D_i)P_i(D_i')$, реализующий часть функциональности алгоритма, обрабатывающий и продуцирующий часть данных, специфицированных на его входе и выходе. Совокупность информационно связанных подсистем, образующих алгоритм, реализуют всю его функциональность, обрабатывает все данные, специфицированные на его входе, и, с помощью дополнительных (вспомогательных, промежуточных) данных, продуцируют все данные, специфицированные на его выходе.

Поскольку алгоритм – это Д-оператор, то для решения приведенных задач запишем его, с учетом определения 4, в виде следующей КС:

$$(D^{\text{ИСХ}}, D^{\text{ИСХ\_}R})A(D^{\text{РЕЗ}}) = (D_1^{\text{ИСХ}}, \text{D}_1^{\text{ИСХ\_}R})P_1(D_1^{\text{РЕЗ}}) * \ldots * (D_k^{\text{ИСХ}}, \text{D}_\text{k}^{\text{ИСХ\_}R})P_k(D_k^{\text{РЕЗ}}). \quad (13)$$

Будем полагать, что данные, специфицированные на входе и выходе алгоритма, являются группами данных и рассмотрим процесс их описания на примере исходных данных.

В результате применения к этим данным операции разгруппировки

$$\overline{\cup} D^{\text{ИСХ}} = D_1^{\text{ИСХ}}, \ldots, D_s^{\text{ИСХ}}, \quad (14)$$

разбиваем исходную группу данных на подгруппы, которые необходимо "распределить" между подсистемами. Если s=k и каждая полученная группа данных согласована с функциональностью одной из подсистем, то есть могут обрабатываться ими, то данные соответствующим образом распределяются, а процесс описания данных на этом завершается.

В противном случае, возможно несколько вариантов описания данных.

Если полученные данные не согласуются с функциональностью подсистем, так как степень детализации полученных данных излишняя или s>k, то данные должны быть перегруппированы. Для этого, применив к полученным данным операцию группировки следующим образом:

$$\begin{gathered} \text{D}_{\text{i}_1}^{\text{ИСХ}} \dot{\cup} \ldots \dot{\cup} \text{D}_{\text{i}_s}^{\text{ИСХ}} = \text{D}_1'^{\text{ИСХ}}; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \text{D}_{\text{i}_p}^{\text{ИСХ}} \dot{\cup} \ldots \dot{\cup} \text{D}_{\text{i}_m}^{\text{ИСХ}} = \text{D}_\text{r}'^{\text{ИСХ}}, \end{gathered} \quad (15)$$

где $i_j \in \{1,\ldots,s\}$ (для всех j), получаем r групп данных с произвольным составом таких, что $\text{r} < \text{s}$, то есть $D^{\text{ИСХ}} = D_1'^{\text{ИСХ}}, \ldots, D_r'^{\text{ИСХ}}$. При $r = k$ и согласованности полученных данных с функциональностью подсистем, процесс их описания на этом завершается. При этом перегруппировке может предшествовать разгруппировка полученных групп данных $D_1'^{\text{ИСХ}}, \ldots, D_r'^{\text{ИСХ}}$ или некоторой их части.

Если требуемый результат не получен, то процесс описания данных может быть начат сначала (с разгруппировка данных $D^{\text{ИСХ}}$) с целью получения других групп данных, которые, в свою очередь, будут перегруппировываться.

Если полученные данные не согласуются с функциональностью подсистем, так как степень детализации данных недостаточна или s<k, то данные необходимо подвергнуть дальнейшей разгруппировке, например, с помощью производной операции (4). В результате некоторые группы данных будут отделены

$$_p\overline{\cup}_r(D^{\text{ИСХ}}) = D_1^{\text{ИСХ}}, \ldots, D_p^{\text{ИСХ}}, D'^{\text{ИСХ}}, D_r^{\text{ИСХ}}, \ldots, D_m^{\text{ИСХ}} \quad (16)$$

и, в случае необходимости, перегруппированы, а группа данных $D'^{\text{ИСХ}}$ может быть перегруппирована отдельно. Любая из полученных групп данных или они все могут подвергаться дальнейшей разгруппировке и/или перегруппировке. В любом случае процесс описания данных продолжается до тех пор, пока количество полученных групп данных не будет соответствовать числу подсистем, а функциональность каждой подсистемы не обеспечит обработку специфицированных данных, то есть, не будет согласована с ними.

Совершенно аналогично описываются и согласуются с функциональностью подсистем остальные данные

$$\overline{*} D^{\text{ИСХ\_R}} = D_1^{\text{ИСХ\_R 1}}, \ldots, D_k^{\text{ИСХ\_R}}, \quad (17)$$

$$\overline{*}\, D^{\text{РЕЗ}} = D_1^{\text{РЕЗ}},\ldots, D_k^{\text{РЕЗ}},$$

в результате чего они ($D_i^{\text{РЕЗ}}$ и $D_i^{\text{ИСХ\_R}}$) специфицируются на входе и выходе каждой i-ой подсистемы.

Таким образом, получаем совокупность СД, посредством которых обеспечивается согласованность специфицированных структур данных с числом и функциональность подсистем алгоритма. В общем случае в процессе детализации могут быть получены структуры данных, которые будут рассмотрены на следующем этапе.

Предложенная КС алгоритма (13), очевидно, не полностью отвечает определению 4, так как в ней специфицированы только глобальные данные, в связи с чем, введем понятие локальных данных.

Локальными будем называть данные, которые можно рассматривать как инкапсулированные в Д-операторе и которые, являясь вспомогательными, специфицируются на входе и выходе производных Д-операторов для реализации их функциональности.

К локальным данным i-ой подсистемы отнесем $D_i^{исх}$ – исходные, $D_i^{\text{исх}\_R}$ – вводимые исходные, – вводимые, D_i^W – выводимые. D_i^R

Понятие связи в КС введем посредством следующего определения.

Определение 5. Д-операторы $(D_i)X_i(D_i')$ и $(D_j)X_j(D_j')$ (где i<j), входящие в КС

$$(D)X(D') = (D_1)X_1(D_1')*\ldots*(D_i)X_i(D_i')*(D_{i+1})X_{i+1}(D_{i+1}')*\ldots*(D_j)X_j(D_j')*\ldots*(D_k)X_k(D_k'),$$

прямо связаны, если для них выполняется соотношение $D_i' \cap D_j \neq \varnothing$, а данные ${}_i\vec{D}_j = D_i' \cap D_j$ – прямо связывающие эти Д-операторы или прямые связи. В случае, когда $D_i' \cap D_{i+1} \neq \varnothing$, Д-операторы $(D_i)X_i(D_i')$ и $(D_{i+1})X_{i+1}(D_{i+1}')$ связаны непосредственно, а данные ${}_i\vec{D}_{i+1}$ – непосредственно связывающие эти Д-операторы или непосредственные связи. Все связи Д-оператора $(D_i)X_i(D_i')$ со всеми следующими за ним и всеми предшествующими ему Д-операторами обозначим $\vec{D}_i^{CB} = \bigcup_{j=i+1}^{k} {}_i\vec{D}_j$ и $\bar{D}_i^{CB} = \bigcup_{j=1}^{i-1} {}_j\bar{D}_i$ и назовем правыми и левыми его связями.

Из определения 5 следует, что каждый Д-оператор $(D_i)X_i(D_i')$ может быть связан со всеми следующими за ним Д-операторами и всеми предшествующими Д-операторами. В дальнейшем будем говорить, что некоторое подмножество выходных данных оператора $(D_i)X_i(D_i')$, который является источником этих данных, поступает на входы Д-операторов $(D_{i+1})X_{i+1}(D_{i+1}'),\ldots,(D_k)X_k(D_k')$, которые являются их приемниками. Отметим, введенные связывающие данные на данном этапе являются локальными.

Учитывая введенные локальные и связывающие данные, КС алгоритма запишем в следующем виде

$$\begin{aligned}
&(D^{\text{ИСХ}}, D^{\text{ИСХ_R}})A(D^{\text{РЕЗ}}) = \\
&= (D_1^{\text{ИСХ}}, D_1^{\text{ИСХ_R}}, D_1^{\text{исх}}, D_1^{\text{исх_R}}, D_1^{\text{R}})P_1(D_1^{\text{РЕЗ}}, D_1^{\text{W}}, \vec{D}_1^{\text{СВ}})*\ldots \\
&\ldots*(\bar{D}_i^{\text{СВ}}, D_i^{\text{ИСХ}}, D_i^{\text{ИСХ_R}}, D_i^{\text{исх}}, D_i^{\text{исх_R}}, D_i^{\text{R}})P_i(D_i^{\text{РЕЗ}}, D_i^{\text{W}}, \vec{D}_i^{\text{СВ}})*\ldots \\
&\ldots*(\bar{D}_k^{\text{СВ}}, D_k^{\text{ИСХ}}, D_k^{\text{ИСХ_R}}, D_k^{\text{исх}}, D_k^{\text{исх_R}}, D_k^{\text{R}})P_k(D_k^{\text{РЕЗ}}, D_k^{\text{W}}),
\end{aligned}$$

где, $\bar{D}_1^{CB} = \varnothing$, $\vec{D}_k^{CB} = \varnothing$, так как нет предшествующих и последующих Д-операторов.

В результате выполненных построений алгоритм разбит на подсистемы, соответствующие определению 4, у которых специфицированы глобальные, локальные и связывающие данные. В приведенной КС специфицированы данные для самого общего случая, который на практике обычно не реализуется. Что бы оценить реальное положение вещей остановимся на свойствах специфицированных данных.

Поскольку $D_{i_r}^{\text{ИСХ}} \dot{\cup} \ldots \dot{\cup} D_{i_p}^{\text{ИСХ}} = D_i^{\text{ИСХ}}$, а семейство множеств $D_i^{\text{ИСХ}}$ будем трактовать как элемент множества, для исходных глобальных данных выполняется соотношение

$$D^{\text{ИСХ}} = (\{D_1^{\text{ИСХ}}\} \cup \ldots \cup \{D_k^{\text{ИСХ}}\}). \tag{18}$$

Аналогичные соотношения выполняются для остальных глобальных данных:

$$D^{\text{ИСХ\_R}} = (\{D_1^{\text{ИСХ\_R}}\} \cup \ldots \cup \{D_k^{\text{ИСХ\_R}}\}), \tag{19}$$

$$D^{\text{РЕЗ}} = (\{D_1^{\text{РЕЗ}}\} \cup \ldots \cup \{D_k^{\text{РЕЗ}}\}).$$

Множества $D_j^{ИСХ}$ или $D_j^{ИСХ\_R}$ могут быть пустыми при условии выполнения соотношения (18) и (19), а $D_j^{РЕЗ} \neq \varnothing$, при любом j. Пустыми могут быть: множества данных $\vec{D}_j^{СВ}$ и/или $\vec{D}_j^{СВ}$, если Д-оператор не связан с предшествующими и/или последующими Д-операторами; $D_j^{исх}$ и/или $D_j^{исх\_R}$, если Д-оператор не имеет локальных исходных данных; D_j^{R} и/или D_j^{W}, если Д-оператор не осуществляет операций ввода-вывода.

Теперь перейдем к следующему этапу, на котором разрабатываются алгоритмы подсистем, для чего каждая из подсистема декомпозируется, а специфицированные у неё данные детализуются. В результате следующий уровень описание алгоритма представляет собой совокупность ПС всех подсистем, которую назовем первым слоем алгоритма.

Далее запишем КС всех подсистем, входящих в этот слой, с последовательной нумерацией производных Д-операторов и следующими обозначениями. Полагая, что все специфицированные у всех подсистем (для всех i) данные являются глобальными, множества данных $D_i^{ИСХ} \cup D_i^{исх}$ и $D_i^{ИСХ\_R} \cup D_i^{исх\_R}$ обозначим ${}^1D_i^{ИСХ\_R}$, ${}^1D_i^{ИСХ}$, связывающие данные ${}^1\vec{D}_i^{СВ}$, ${}^1\vec{D}_i^{СВ}$, а подсистему – ${}^1P_i$, где левый верхний индекс указывает уровень (этап) детализации алгоритма. В качестве локальных данных, аналогично вышерассмотренному случаю, будем использовать ${}^2D_i^{исх}$ – исходные, ${}^2D_i^{исх\_R}$ – вводимые исходные, ${}^2D_i^{r}$ – вводимые, ${}^2D_i^{w}$ – выводимые и связывающие данные ${}^2\vec{D}_i^{св}$, ${}^2\vec{D}_i^{св}$.

$$({}^1D_1^{ИСХ}, {}^1D_1^{ИСХ\_R}, {}^1D_1^{R})\,{}^1P_1({}^1D_1^{РЕЗ}, {}^1D_1^{W}, {}^1\vec{D}_1^{СВ}) =$$
$$= ({}^2D_1^{ИСХ}, {}^2D_1^{ИСХ\_R}, {}^2D_1^{исх}, {}^2D_1^{исх\_R}, {}^2D_1^{R}, {}^2D_1^{r})\,{}^2X_1({}^2D_1^{РЕЗ}, {}^2D_1^{W}, {}^2D_1^{w}, {}^2\vec{D}_1^{СВ}, {}^2\vec{D}_i^{св}) * \ldots$$
$$\ldots * ({}^2\vec{D}_i^{св}, {}^2D_i^{ИСХ}, {}^2D_i^{ИСХ\_R}, {}^2D_i^{исх}, {}^2D_i^{исх\_R}, {}^2D_i^{R}, {}^2D_i^{r})\,{}^2X_i({}^2D_i^{РЕЗ}, {}^2D_i^{W}, {}^2D_i^{w}, {}^2\vec{D}_i^{СВ}, {}^2\vec{D}_i^{св}) * \ldots$$
$$\ldots * ({}^2\vec{D}_{m_1}^{св}, {}^2D_{m_1}^{ИСХ}, {}^2D_{m_1}^{ИСХ\_R}, {}^2D_{m_1}^{исх}, {}^2D_{m_1}^{исх\_R}, {}^2D_{m_1}^{R}, {}^2D_{m_1}^{r})\,{}^2X_{m_1}({}^2D_{m_1}^{РЕЗ}, {}^2D_{m_1}^{W}, {}^2\vec{D}_{m_1}^{СВ})$$

$$({}^1\vec{D}_2^{СВ}, {}^1D_2^{ИСХ}, {}^1D_2^{ИСХ\_R}, {}^1D_2^{R})\,{}^1P_2({}^1D_2^{РЕЗ}, {}^1D_2^{W}, {}^1\vec{D}_2^{СВ}) =$$
$$= ({}^2\vec{D}_{m_1+1}^{СВ}, {}^2D_{m_1+1}^{ИСХ}, {}^2D_{m_1+1}^{ИСХ\_R}, {}^2D_{m_1+1}^{исх}, {}^2D_{m_1+1}^{исх\_R}, {}^2D_{m_1+1}^{R}, {}^2D_{m_1+1}^{r})\,{}^2X_{m_1+1}({}^2D_{m_1+1}^{РЕЗ}, {}^2D_{m_1+1}^{W}, {}^2D_{m_1+1}^{w}, {}^2\vec{D}_{m_1+1}^{СВ}, {}^2\vec{D}_{m_1+1}^{св}) * \ldots$$
$$\ldots * ({}^2\vec{D}_{m_1+i}^{СВ}, {}^2\vec{D}_{m_1+i}^{св}, {}^2D_{m_1+i}^{ИСХ}, {}^2D_{m_1+i}^{ИСХ\_R}, {}^2D_{m_1+i}^{исх}, {}^2D_{m_1+i}^{исх\_R}, {}^2D_{m_1+i}^{R}, {}^2D_{m_1+1}^{r})\,{}^2X_{m_1+i} \times$$
$$\times ({}^2D_{m_1+i}^{РЕЗ}, {}^2D_{m_1+i}^{W}, {}^2D_{m_1+i}^{w}, {}^2\vec{D}_{m_1+i}^{СВ}, {}^2\vec{D}_{m_1+i}^{св}) * \ldots$$
$$\ldots * ({}^2\vec{D}_{m_2}^{СВ}, {}^2\vec{D}_{m_2}^{св}, {}^2D_{m_2}^{ИСХ}, {}^2D_{m_2}^{ИСХ\_R}, {}^2D_{m_2}^{исх}, {}^2D_{m_2}^{исх\_R}, {}^2D_{m_2}^{R}, {}^2D_{m_2}^{r})\,{}^2X_{m_2}({}^2D_{m_2}^{РЕЗ}, {}^2D_{m_2}^{W}, {}^2D_{m_2}^{w}, {}^2\vec{D}_{m_2}^{СВ})$$

……

$$({}^1\vec{D}_i^{СВ}, {}^1D_i^{ИСХ}, {}^1D_i^{ИСХ\_R}, {}^1D_i^{R})\,{}^1P_i({}^1D_i^{РЕЗ}, {}^1D_i^{W}, {}^1\vec{D}_i^{СВ}) =$$
$$= ({}^2\vec{D}_{m_{i-1}+1}^{СВ}, {}^2D_{m_{i-1}+1}^{ИСХ}, {}^2D_{m_{i-1}+1}^{ИСХ\_R}, {}^2D_{m_{i-1}+1}^{исх}, {}^2D_{m_{i-1}+1}^{исх\_R}, {}^2D_{m_{i-1}+1}^{R}, {}^2D_{m_{i-1}+1}^{r})\,{}^2X_{m_{i-1}+1} \times$$
$$\times ({}^2D_{m_{i-1}+1}^{РЕЗ}, {}^2D_{m_{i-1}+1}^{W}, {}^2D_{m_{i-1}+1}^{w}, {}^2\vec{D}_{m_{i-1}+1}^{СВ}, {}^2\vec{D}_{m_{i-1}+1}^{св}) * \ldots$$
$$\ldots * ({}^2\vec{D}_{m_{i-1}+i}^{СВ}, {}^2\vec{D}_{m_{i-1}+i}^{св}, {}^2D_{m_{i-1}+i}^{ИСХ}, {}^2D_{m_{i-1}+i}^{ИСХ\_R}, {}^2D_{m_{i-1}+i}^{исх}, {}^2D_{m_{i-1}+i}^{исх\_R}, {}^2D_{m_{i-1}+i}^{R}, {}^2D_{m_{i-1}+i}^{r})\,{}^2X_{m_{i-1}+i} \times$$
$$\times ({}^2D_{m_{i-1}+i}^{РЕЗ}, {}^2D_{m_{i-1}+i}^{W}, {}^2D_{m_{i-1}+i}^{w}, {}^2\vec{D}_{m_{i-1}+i}^{СВ}, {}^2\vec{D}_{m_{i-1}+i}^{св}) * \ldots$$
$$\ldots * ({}^2\vec{D}_{m_i}^{СВ}, {}^2\vec{D}_{m_i}^{св}, {}^2D_{m_i}^{ИСХ}, {}^2D_{m_i}^{ИСХ\_R}, {}^2D_{m_i}^{исх}, {}^2D_{m_i}^{исх\_R}, {}^2D_{m_i}^{R}, {}^2D_{m_i}^{r})\,{}^2X_{m_i}({}^2D_{m_i}^{РЕЗ}, {}^2D_{m_i}^{W}, {}^2D_{m_i}^{w}, {}^2\vec{D}_{m_i}^{СВ})$$

……

$$({}^1\vec{D}_k^{СВ}, {}^1D_k^{ИСХ}, {}^1D_k^{ИСХ\_R}, {}^1D_k^{R})\,{}^1P_k({}^1D_k^{РЕЗ}, {}^1D_k^{W}) =$$
$$= ({}^2\vec{D}_{m_{k-1}+1}^{СВ}, {}^2D_{m_{k-1}+1}^{ИСХ}, {}^2D_{m_{k-1}+1}^{ИСХ\_R}, {}^2D_{m_{k-1}+1}^{исх}, {}^2D_{m_{k-1}+1}^{исх\_R}, {}^2D_{m_{k-1}+1}^{R}, {}^2D_{m_{k-1}+1}^{r})\,{}^2X_{m_{k-1}+1} \times$$
$$\times ({}^2D_{m_{k-1}+1}^{РЕЗ}, {}^2D_{m_{k-1}+1}^{W}, {}^2D_{m_{k-1}+1}^{w}, {}^2\vec{D}_{m_{k-1}+1}^{св}) * \ldots$$
$$\ldots * ({}^2\vec{D}_{m_{k-1}+i}^{СВ}, {}^2\vec{D}_{m_{k-1}+i}^{св}, {}^2D_{m_{k-1}+i}^{ИСХ}, {}^2D_{m_{k-1}+i}^{ИСХ\_R}, {}^2D_{m_{k-1}+i}^{исх}, {}^2D_{m_{k-1}+i}^{исх\_R}, {}^2D_{m_{k-1}+i}^{R}, {}^2D_{m_{k-1}+i}^{r})\,{}^2X_{m_{k-1}+i} \times$$
$$\times ({}^2D_{m_{k-1}+i}^{РЕЗ}, {}^2D_{m_{k-1}+i}^{W}, {}^2D_{m_{k-1}+i}^{w}, {}^2\vec{D}_{m_{k-1}+i}^{св}) * \ldots$$
$$\ldots * ({}^2\vec{D}_{m_k}^{СВ}, {}^2\vec{D}_{m_k}^{св}, {}^2D_{m_k}^{ИСХ}, {}^2D_{m_k}^{ИСХ\_R}, {}^2D_{m_k}^{исх}, {}^2D_{m_k}^{исх\_R}, {}^2D_{m_k}^{R}, {}^2D_{m_k}^{r})\,{}^2X_{m_k}({}^2D_{m_k}^{РЕЗ}, {}^2D_{m_k}^{W}, {}^2D_{m_k}^{w}).$$

Описание данных рассмотрим на примере некоторой j-ой подсистемы.

Группы данных детализуются аналогично вышерассмотренному случаю (см. (14) – (17)) и записываются в виде СД. Если в результате детализации будет получена структура данных (запись, многомерный или одномерный массив записей, многомерный или одномерный массив), то она детализуется посредством операций (2) – (6), (8) – (12). Например, при разгруппировании ${}^1D_j^{\text{ИСХ}}$ получили

$$\dot{\cup}({}^1D_j^{\text{ИСХ}})={}^2D_1^{\text{ИСХ}},...,{}^2D_r^{\text{ИСХ\_ЗАП}},...,{}^2D_s^{\text{ИСХ}},$$

где ${}^2D_r^{\text{ИСХ\_ЗАП}}$ – запись. Выполнив дальнейшую детализацию записи,

$$\dot{*}({}^2\mathrm{D}_{\mathrm{r}}^{\text{ИСХ\_ЗАП}})={}^2\mathrm{D}_{\mathrm{r_1}}^{\text{ИСХ\_ЗАП}},...,{}^2\mathrm{D}_{\mathrm{r_p}}^{\text{ИСХ\_ЗАП}}$$

получаем записи предшествующего уровня вложенности, которые могут быть перегруппированы, в результате чего могут быть включены (все или их часть) в новые группы специфицируемых данных. Эти полученные группы данных и/или полученные записи (перегруппированные, в случае необходимости) распределяются между производными Д-операторами.

Поскольку для записей выполняется соотношение

$${}^2D_{r_1}^{\text{ИСХ\_ЗАП}} * ... * {}^2D_{r_p}^{\text{ИСХ\_ЗАП}}={}^2D_r^{\text{ИСХ\_ЗАП}},$$

а для исходных данных

$${}^2D_1^{\text{ИСХ}} \dot{\cup} ... \dot{\cup} {}^2D_r^{\text{ИСХ\_ЗАП}} \dot{\cup} ... \dot{\cup} {}^2D_s^{\text{ИСХ}}={}^1D_j^{\text{ИСХ}},$$

то семейство множеств ${}^2D_p^{\text{ИСХ}}$ и упорядоченное семейство множеств ${}^2D_r^{\text{ИСХ\_ЗАП}}$ могут трактоваться как элементы множества. Поскольку так же могут детализоваться и остальные глобальные данные, специфицированные в КС, то для них выполняются соотношения (аналогичные (18) – (19)):

$${}^1D_{\mathrm{j}}^{\text{ИСХ}} = \bigcup_{i=m_{j-1}+1}^{m_j}\{{}^2D_{\mathrm{i}}^{\text{исх}}\}\ ; \quad {}^1D_{\mathrm{j}}^{\text{ИСХ\_R}} = \bigcup_{i=m_{j-1}+1}^{m_j}\{{}^2D_{\mathrm{i}}^{\text{ИСХ\_R}}\}\ ; \ {}^1D_{\mathrm{j}}^{\mathrm{R}} = \bigcup_{i=m_{j-1}+1}^{m_j}\{{}^2D_{\mathrm{i}}^{\mathrm{R}}\}\ ; \quad {}^1\mathrm{D}_{\mathrm{j}}^{\mathrm{W}} = \bigcup_{\mathrm{i}=\mathrm{m_{j-1}}+1}^{\mathrm{m_j}}\{{}^2\mathrm{D}_{\mathrm{i}}^{\mathrm{W}}\}\ ;$$

$${}^1\mathrm{D}_{\mathrm{j}}^{\text{РЕЗ}} = \bigcup_{\mathrm{i}=\mathrm{m_{j-1}}+1}^{\mathrm{m_j}}\{{}^2\mathrm{D}_{\mathrm{i}}^{\text{РЕЗ}}\}\ ; \ {}^1\vec{\mathrm{D}}_{\mathrm{j}}^{\text{СВ}} = \bigcup_{\mathrm{i}=\mathrm{m_{j-1}}+1}^{\mathrm{m_j}}\{{}^2\vec{\mathrm{D}}_{\mathrm{i}}^{\text{СВ}}\}\ ; \ {}^1\vec{\mathrm{D}}_{\mathrm{j}}^{\text{СВ}} = \bigcup_{\mathrm{i}=\mathrm{m_{j-1}}+1}^{\mathrm{m_j}}\{{}^2\vec{\mathrm{D}}_{\mathrm{i}}^{\text{СВ}}\}\ ,$$

при условии выполнения которых множества данных

$${}^2\vec{D}_{\mathrm{p}}^{\text{СВ}}, {}^2D_{\mathrm{p}}^{\text{ИСХ}}, {}^2D_{\mathrm{p}}^{\text{ИСХ\_R}}, {}^2D_{\mathrm{p}}^{\mathrm{R}}, {}^2D_{\mathrm{p}}^{\text{РЕЗ}}, {}^2D_{\mathrm{p}}^{\mathrm{W}}, {}^2\vec{D}_{\mathrm{p}}^{\text{СВ}},$$

где $p \in \{1,...,m_k\}$, могут быть пустыми. Кроме того, ${}^2\vec{D}_1^{CB},...,{}^2\vec{D}_{m_1}^{CB}=\varnothing$; ${}^2\vec{D}_{m_{k-1}+1}^{CB},...,{}^2\vec{D}_{m_k}^{CB}=\varnothing$ и ${}^2\vec{D}_1^{\text{св}}$, ${}^2\vec{D}_{m_1+1}^{\text{св}},...$ …, ${}^2\vec{D}_{m_{k-1}+1}^{\text{св}}=\varnothing$; ${}^2\vec{D}_{m_1}^{\text{св}},\ldots,{}^2\vec{D}_{m_k}^{\text{св}}=\varnothing$.

Для глобальных данных алгоритма в этом слое выполняются следующие соотношения:

$$D^{\text{ИСХ}} \subseteq (\{{}^2D_1^{\text{ИСХ}}\} \cup ... \cup \{{}^2D_{\mathrm{m_k}}^{\text{ИСХ}}\});$$

$$D^{\text{ИСХ\_R}} \subseteq (\{{}^2D_1^{\text{ИСХ\_R}}\} \cup ... \cup \{{}^2D_{\mathrm{m_k}}^{\text{ИСХ\_R}}\});$$

$$D^{\text{РЕЗ}} = (\{{}^2D_1^{\text{РЕЗ}}\} \cup ... \cup \{{}^2D_{\mathrm{m_k}}^{\text{РЕЗ}}\}).$$

По поводу связывающих данных будем утверждать следующее.

Утверждение. В данном слое алгоритма для глобальных связывающих данных выполняется соотношение

$$\bigcup_{j=1}^{m_{k-1}} {}^2\vec{D}_j^{CB} = \bigcup_{p=2}^{m_k} {}^2\vec{D}_p^{CB},$$

а для локальных связывающих данных в каждой КС – соотношения: и

$$\bigcup_{j=m_{i-1}+1}^{m_i-1} {}^2\vec{D}_j^{св} = \bigcup_{p=m_{i-1}+2}^{m_i} {}^2\bar{D}_p^{св} .$$

Доказательство очевидно и следует из определения 5.

На основе этого утверждения, покажем возможность спецификации информационных связей в слое алгоритма и КС, детализовав связывающие данные в виде:

$${}^2\vec{D}_j = {}^2_j\vec{D}_{j+1},...,{}^2_j\vec{D}_p\ ,\ {}^2\bar{D}_j = {}^2_1\bar{D}_j,...,{}^2_{j-1}\bar{D}_j\ ,$$

где левый и правый нижний индекс соответствуют источнику и приемнику данных, и, обозначив для краткости данные, отличные от связывающих, D и D'.

С учетом введенных обозначений слой алгоритма для случая, когда все подсистемы и все КС, входящие в подсистемы, связаны, запишем следующим образом:

$$(D)^1P_1({}^1_1\vec{D}_2^{\text{CB}},...,{}^1_1\vec{D}_{\text{k}}^{\text{CB}}) = (D)^2X_1({}^2_1\vec{D}_{m_1+1}^{\text{CB}},...,{}^2_1\vec{D}_{m_{i-1}+1}^{\text{CB}},...,{}^2_1\vec{D}_{\text{m}_\text{k}}^{\text{CB}},{}^2_1\vec{D}_2^{\text{св}},...,{}^2_1\vec{D}_{\text{m}_1}^{\text{св}})*...$$
$$...*({}^2_1\vec{D}_{\text{i}}^{\text{св}},...,{}^2_{i-1}\vec{D}_{\text{i}}^{\text{св}})^2X_i({}^2_i\vec{D}_{m_1+1}^{\text{CB}},...,{}^2_i\vec{D}_{\text{m}_\text{k}}^{\text{CB}},{}^2_i\vec{D}_{\text{i}+1}^{\text{св}},...,{}^2_i\vec{D}_{\text{m}_1}^{\text{св}})*({}^2_1\vec{D}_{m_1}^{\text{св}},...,{}^2_{m_1-1}\vec{D}_{m_1}^{\text{св}})^2X_{m_1}({}^2_{m_1}\vec{D}_{m_1+1}^{\text{CB}},...,{}^2_{m_1}\vec{D}_{m_k}^{\text{CB}})$$

……

$$({}_1\vec{D}_{\text{i}}^{\text{CB}},{}_2\vec{D}_{\text{i}}^{\text{CB}},...,{}_{i-1}\vec{D}_{\text{i}}^{\text{CB}})P_i({}_i\vec{D}_{\text{i}+1}^{\text{CB}},...,{}_i\vec{D}_{\text{k}}^{\text{CB}}) =$$
$$=({}^2_1\vec{D}_{m_{i-1}+1}^{\text{CB}},...,{}^2_{m_{i-1}}\vec{D}_{m_{i-1}+1}^{\text{CB}})^2X_{m_{i-1}+1}({}^2_{m_{i-1}+1}\vec{D}_{m_i+1}^{\text{CB}},...,{}^2_{m_{i-1}+1}\vec{D}_{m_k}^{\text{CB}},{}^2_{m_{i-1}+1}\vec{D}_{m_{i-1}+2}^{\text{св}},...,{}^2_{m_{i-1}+1}\vec{D}_{\text{m}_\text{i}}^{\text{св}})*...$$
$$...*({}^2_{m_{i-1}+1}\vec{D}_{m_{i-1}+i}^{\text{св}},...,{}^2_{m_{i-1}+(i-1)}\vec{D}_{m_{i-1}+i}^{\text{св}})^2X_{m_{i-1}+i}({}^2_{m_{i-1}+i}\vec{D}_{m_i+1}^{\text{CB}},...,{}^2_{m_{i-1}+i}\vec{D}_{m_k}^{\text{CB}},{}^2_{m_{i-1}+i}\vec{D}_{m_{i-1}+(i+1)}^{\text{св}},...,{}^2_{m_{i-1}+i}\vec{D}_{\text{m}_\text{i}}^{\text{св}})*...$$
$$...*({}^2_1\vec{D}_{m_i}^{\text{св}},...,{}^2_{m_i-1}\vec{D}_{m_i}^{\text{св}})^2X_{m_i}({}^2_{m_i}\vec{D}_{m_i+1}^{\text{CB}},...,{}^2_{m_i}\vec{D}_{\text{k}}^{\text{CB}})$$

…. ……..

$$({}_1\vec{D}_{\text{k}}^{\text{CB}},{}_2\vec{D}_{\text{k}}^{\text{CB}},...,{}_{k-1}\vec{D}_{\text{k}}^{\text{CB}})P_k(D') = ({}^2_1\vec{D}_{m_{k-1}+1}^{\text{CB}},...,{}^2_{m_{k-1}}\vec{D}_{m_{k-1}+1}^{\text{CB}})^2X_{m_{k-1}+1}({}^2_{m_{k-1}+1}\vec{D}_{m_{k-1}+2}^{\text{св}},...,{}^2_{m_{k-1}+1}\vec{D}_{\text{m}_\text{k}}^{\text{св}})*...$$
$$...*({}^2_1\vec{D}_{m_{k-1}+i}^{\text{св}},...,{}^2_{m_{k-1}+(i-1)}\vec{D}_{m_{k-1}+i}^{\text{св}})^2X_{m_{k-1}+i}({}^2_{m_{k-1}+i}\vec{D}_{m_{k-1}+(i+1)}^{\text{св}},...,{}^2_{m_{k-1}+i}\vec{D}_{\text{m}_\text{k}}^{\text{св}})*...$$
$$...*({}^2_1\vec{D}_{m_k}^{\text{св}},...,{}^2_{m_k-1}\vec{D}_{m_i}^{\text{св}})^2X_{m_k}(D').$$

В предложенном варианте записи слоя алгоритма специфицированы не только все возможные связи, а источники и приемники данных поставлены в однозначное соответствие.

Заключение

На описанном этапе разработки поставленная задача решена. Средствами рассматриваемого алгебраического аппарата согласованно описаны три важнейших аспекта описания алгоритма: поток управления, обрабатываемые данные и информационные связи. В процессе решения указанной задачи выявлены свойства данных, из которых видно, что все специфицированные данные обрабатываются, а подлежащие обработке специфицируются.

Этот результат естественно переносится и на все следующие этапы, состоящие в повторении показанных действий при сохранении свойств специфицируемых данных. Отличие каждого следующего шага (от предыдущего) состоит в получении все более "толстых" слоев алгоритма и замене Д-операторов соответствующей функциональности операциями алгебры. После такой замены декомпозиции подвергаются Д-операторы, входящие в такую операцию.

Описанный процесс продолжается до достижения такого уровня детализации, после которого возможен переход от алгоритма к программе, что показано в работе [8].

Из изложенного легко увидеть, что указанная задача решается в рамках нисходящей стратегии проектирования алгоритмов. Доказанная в работе [9] теорема о возможности объединения Д-операторов позволяет, проведя рассуждения в порядке обратном вышеприведенному, получить аналогичный результат для восходящей стратегии проектирования алгоритмов и программ.

1. *Данные* в языках программирования: абстракция и типология. Сб. статей / Под ред. В. Агафонова. – М.: Мир, 1982. – 328 с.
2. *Bastani F.B., Iyengar S.S.* The effect of data structures on the logical complexity of programs // CACM. – 1987. – V. 30, N 3. – P. 250–259.
3. *Акуловский В.Г.* Основы алгебры алгоритмов, базирующейся на данных // Проблеми програмування. – 2010. – № 2–3. – С.89–96.
4. *Акуловский В.Г.* Алгебра алгоритмов, базирующаяся на данных // Кибернетика и системный анализ. – 2012. – № 2. – С. 151–166.
5. *Дорошенко А.Е., Акуловский В.Г.* Алгебра алгоритмов с данными и прогнозирование вычислительного процесса // Проблеми програмування. – 2011. – № 3. – С. 3–10.

6. *Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л.* Алгебра. Языки. Программирование. – К.: Наукова думка, 1978. – 319 с.
7. *Акуловский В.Г. , Дорошенко А.Е.* Нисходящее проектирование алгоритмов в рамках алгеброалгоритмического подхода // Математические машины и системы. – 2012. – № 3. – С. 97–102.
8. *Акуловский В.Г.* Реализация формализованного перехода от алгоритма к программе средствами расширенной алгебры алгоритмов // Проблеми програмування. – 2008. – № 1. – С. 51–59.
9. *Дорошенко А.Ю., Акуловський В.Г.* Висхідне проектування алгоритмів при алгеброалгоритмичному підході // Вісник Київського національного університету імені Тараса Шевченка. Серія фізико-математичні науки, 2012. – Випуск 1. – С. 167–172.