УДК 519.683

В.Г. Акуловский

НЕКОТОРЫЕ АСПЕКТЫ ФОРМАЛИЗАЦИИ ДАННЫХ И ДЕКОМПОЗИЦИЯ Д-ОПЕРАТОРОВ

В рамках расширенной алгебры алгоритмов предложен подход к формализации данных. Рассмотрены свойства формализованных данных и некоторые аспекты декомпозиции Д-операторов.

Введение

Принципиально важная роль данных в программировании давно осознана программистским сообществом (см. например, [1]), что нашло своё выражение в формализации данных как для компиляции программ, так и для разработки алгоритмов. Учитывая, что возможности формализации данных в алгоритмике реализованы далеко не полностью. В то же время решение этой задачи позволяет рассчитывать на формализацию некоторых аспектов декомпозиции операторов, преобразование алгоритмов, контроль их корректности и т.д. на достаточно ранних стадиях разработки алгоритмов.

Предпосылкой и основой для выполнения данной работы послужила мысль, высказанная в [2], о том, что в известной модели ЭВМ В.М. Глушкова [3, 4] в качестве множества состояний операционного автомата могут рассматриваться обрабатываемые алгоритмом данные.

Некоторым аспектам формализации данных в рамках расширенной алгебры алгоритмов [5], в которую введено понятие взаимосвязи операторов и данных [6], рассмотрены автором в [7, 8] и ранее полученные результаты в данном случае будут уточняться и развиваться.

Цель данной работы – разработка подхода к формализации данных на этапе декомпозиции операторов, из которых строится алгоритм в рамках алгебры алгоритмов.

Формализация данных и Д-операторов

Исходя из предпосылки, сформулированной во введении, будем полагать, что информационное множество (множества состояний операционного автомата) представляет собой множество данных Δ.

Для того чтобы определить понятие данные, будем рассматривать их с той точки зрения, что это совокупности ячеек памяти D_i, представляющие собой множество $\Delta = \bigcup_{i=1}^{k} D_i$, которые могут принимать некоторые множества значения ${}_{D_j}I$ из конечного множества значений $\Psi = \bigcup_{i=1}^{k} {}_{D_i}I$. Исходя из этого, определим данные таким образом.

Определение 1. Данными будем называть такое множество ячеек памяти, для которых выполняется соотношение $D_j \sim_{D_j} I$ ($D_j \subseteq \Delta$, ${}_{D_j}I \subseteq \Psi$), т. е. такое множество ячеек памяти D_j, которому в однозначное соответствие поставлено множество значений ${}_{D_j}I$. При этом одновременно любому D_j во взаимно однозначное соответствие может быть поставлено одно и только одно множество ${}_{D_j}I_l$. При $D_j = \varnothing$, будем считать, что ${}_{D_j}I = \varnothing$.

Теперь, исходя из определения 1, введем понятие Д-оператора, который изменяет состояние операционного автомата.

Определение 2. Д-оператор $(D)A(D')$ преобразует множество входных данных $D \subseteq \Delta$, которому в однозначное соответствие было поставлено множество значений ${}_DI_l \subseteq {}_DI$ ($D \sim_D I_l$), в множество выходных данных $D' \subseteq \Delta$, которому во взаимно однозначное соответствие ставится множество значений ${}_{D'}I_p \subseteq {}_{D'}I$ ($D' \sim_{D'} I_p$). Множества входных D и выходных D' данных могут быть пустыми.

Учитывая, что множества входных и выходных данных могут быть пустыми, Д-оператор вида $(\varnothing)A(D')$ будем называть генерирующим, вида $(D)A(\varnothing)$ – неопределенным, а вида $(\varnothing)A(\varnothing)$ – тождественным.

Для дальнейшей формализации определим структуру входных и выходных данных Д-операторов.

Определение 3. Представим входные и выходные данные Д – оператора $(D)A(D')$ в виде следующих подмножеств:

- $D = \hat{D} \cup \widehat{D}$, таких, что $\hat{D} \cap \widehat{D} = \varnothing$ и $\widehat{D} \cap D' = \varnothing$;

- $D' = \hat{D}' \cup \widetilde{D}'$, таких, что $\hat{D}' \cap \widetilde{D}' = \varnothing$ и $\widetilde{D}' \cap D = \varnothing$,

которые назовем: $\widehat{D}$ – исходные, $\hat{D}$, $\hat{D}'$ – проходные, $\widetilde{D}$ – производные. При этом, $D \cap D' = \hat{D} = \hat{D}'$. Любое из подмножеств, образующих множества D и D' (как и сами эти множества) может быть пустым.

Из определения следуют приведенные далее свойства Д-операторов.

Исходные данные не появляются на выходе Д-оператора и, таким образом, не изменяются, производные данные не являются входными, а продуцируются Д-оператором и входят в состав выходных, а множества проходных данных $\hat{D} \subseteq D$ и $\hat{D}' \subseteq D'$ представляют собой одно множество ячеек памяти, по-разному обозначенное на входе и выходе Д-оператора. То есть, эти данные, присутствующие на входе, изменяются, после чего входят в состав выходных. Для множеств D и D' допустимо любое из следующих соотношений:

$$D \subset D', \quad D' \subset D, \quad D = D', \quad D \cap D' = \varnothing, \quad D \cap D' \neq \varnothing.$$

Исходя из того, что множества входных и выходных данных могут быть пустыми и могут пересекаться, приведем следующие возможные варианты Д-оператора:

$$D \cap D' = \varnothing \;\rightarrow\; (\widehat{D})A(\widetilde{D}'); \qquad (1)$$

$$D \cap D' = \varnothing, \widehat{D} = \varnothing \;\rightarrow\; (\varnothing)A(\widetilde{D}'); \qquad (2)$$

$$D = D' \;\rightarrow\; (\hat{D})A(\hat{D}'); \qquad (3)$$

$$D \supset D' \;\rightarrow\; (\widehat{D}, \hat{D})A(\hat{D}'); \qquad (4)$$

$$D \subset D' \;\rightarrow\; (\hat{D})A(\widetilde{D}', \hat{D}'); \qquad (5)$$

$$D \cap D' \neq \varnothing \;\rightarrow\; (\widehat{D}, \hat{D})A(\widetilde{D}', \hat{D}'). \qquad (6)$$

Для того чтобы определить общий случай тождественного Д-оператора, введем понятие неидентичных и идентичных данных.

Определение 4. Множества данных D_i и D_j неидентичны ($D_i >< D_j$), если $D_i \sim_{D_i} I$ и $D_j \sim_{D_j} I$ такие, что $D_i \neq D_j$ или/и ${}_{D_i}I \neq {}_{D_j}I$. И множества данных D_i и D_j идентичны ($D_i <> D_j$), если $D_i \sim_{D_i} I$ и $D_j \sim_{D_j} I$ такие, что $D_i = D_j$ и ${}_{D_i}I = {}_{D_j}I$. В частном случае, когда $D_i = D_j = \varnothing$, они также идентичны.

То есть, множества данных неидентичны, если это различные множества ячеек памяти или, если одно и то же множество ячеек принимает различные значения. Данные идентичны, если одно и то же множество ячеек принимает одинаковые значения.

Исходя из определения 2 – 4, введем понятие тождественного Д-оператора, обозначив $\widetilde{D}$ – множество производных данных до выполнения оператора $(D)A(D')$, а $\widetilde{D}'$ – то же множество данных после его выполнения.

Определение 5. Тождественным назовем Д-оператор $(D)A(D')$, если для множеств, образующих выходные данные D' этого оператора, выполняется одно из следующих пар соотношений:

$\hat{D}' <> \hat{D}$ и $\widetilde{D}' <> \widetilde{D}$;

$\hat{D}' <> \hat{D}$ и $\widetilde{D}' = \varnothing$;

$\hat{D}' = \hat{D} = \varnothing$ и $\widetilde{D}' <> \widetilde{D}$;

$\hat{D}' = \hat{D} = \varnothing$ и $\widetilde{D}' = \varnothing$.

То есть, множество проходных данных на входе Д-оператора идентично множеству проходных данных на его выходе, а множество производных данных пусто или не изменяется в результате выполнения Д-оператора. Проще говоря, Д-оператор тождественный, если он не изменяет обрабатываемых данных. В соответствии с определением 5, тождественным является и Д-оператор $(\varnothing)A(\varnothing)$, что согласуется с определением 2.

Докажем, что в рамках предлагаемого подхода Д-оператор и обрабатываемые им данные обладают некоторыми важными свойствами.

Теорема 1. Д-оператор $(D)A(D')$, если он нетождественный, либо преобразует (изменяет) проходные данные, то есть $\hat{D}><\hat{D}'$, либо продуцирует на своем выходе производные данные, т. е. $\tilde{D}'><\tilde{D}$, либо выполняет оба действия одновременно.

Доказательство построим от противного и рассмотрим три возможных случая, вытекающие из приведенных вариантов Д-оператора (1) – (6).

Первый случай, когда Д-оператор продуцирует производные данные, а проходные данные отсутствуют $\hat{D}=\hat{D}'=\varnothing$, имеет место, когда выполняются соотношения (1) и (2). Предположив, что Д-оператор не генерирует производных данных, т. е. $\tilde{D}'<>\tilde{D}$, то, в соответствии с определением 5, получим тождественный Д-оператор, что противоречит условиям теоремы. Таким образом, Д-оператор в данном случае продуцирует производные данные, т. е. $\tilde{D}'><\tilde{D}$.

Второй случай, когда Д-оператор не продуцирует производных данных $\tilde{D}'=\varnothing$, а обрабатывает только проходные данные, имеет место, когда выполняются соотношения (3) и (4). Предположив, что проходные данные не изменяются, т. е. $\hat{D}<>\hat{D}'$, в соответствии с определением 5, получим тождественный Д-оператор, что противоречит условиям теоремы. Таким образом, Д-оператор в данном случае изменяет проходные данные, т. е. $\hat{D}><\hat{D}'$.

Третий случай, когда Д-оператором продуцируются производные, а на входе и выходе присутствуют проходные данные, имеет место, когда выполняются соотношения (5) и (6).

Предположим, что не изменяются ни проходные, ни производные данные, т. е., предположим, что выполняются соотношения $\hat{D}<>\hat{D}'$ и $\tilde{D}'<>\tilde{D}$. В этом случае, в соответствии с определением 5, получим тождественный Д-оператор, что противоречит условиям теоремы. Д-оператор не будет тождественным в случае, когда выполняется $\hat{D}><\hat{D}'$ и/или $\tilde{D}'><\tilde{D}$, т. е., изменяются проходные или/и производные данные.

Теорема доказана.

Исходя из доказанной теоремы, уместно сделать следующее существенное замечание.

Различия в обозначении множества проходных данных $\hat{D}$ на входе и $\hat{D}'$ на выходе Д-оператора, использованное в определении 3, обуславливается тем, что они неидентичны, если Д-оператор нетождественный.

Определив данные, Д-операторы и некоторые их свойства, перейдем к рассмотрению декомпозиции Д-операторов.

Декомпозиция Д-операторов

Начнем с определения операции композиции, входящей в сигнатуру операций алгебры алгоритмов.

Определение 6. Композиция Д-операторов $(D_B)B(D'_B)*(D_C)C(D'_C)$ означает последовательное выполнение сначала Д-оператора $(D_B)B(D'_B)$, а затем Д-оператора $(D_C)C(D'_C)$. То есть

$$(D_B)B(D'_B)*(D_C)C(D'_C)=$$
$$=((D_B)B(D'_B)\cup D_C)C(D'_C\cup D'_B).$$

Поскольку основным подходом к разработке алгоритмов является декомпозиция образующих его операторов, введем, основываясь на определении 6, следующую аксиому.

Аксиома. Любой Д-оператор, за исключением элементарных, может быть представлен в виде композиции двух других Д-операторов

$$(D_A)A(D'_A)=(D_B)B(D'_B)*(D_C)C(D'_C),$$

которые в общем случае обладают следующими свойствами:

$$D_B \subseteq D_A,\ D'_C \subseteq D'_A,$$

$$D_C \cap D_A \neq \varnothing,\ D'_B \cap D'_A \neq \varnothing.$$

Д-оператор $(D_A)A(D'_A)$ будем называть исходным, а $(D_B)B(D'_B)$ и $(D_C)C(D'_C)$ – производными.

Трактовка декомпозиции Д-операторов, данная в этой аксиоме, допустимая на некотором (достаточно высоком) уровне абстракции, не позволяет детализовать рассмотрение этой принципиально важной операции, так как не учитывается то факт, что производные Д-операторы совместно решают одну общую для них задачу, адекватную задаче, решаемой исходным Д-оператором.

Перечислим те аспекты декомпозиции, которые необходимо реализовать для того, чтобы учесть упомянутый факт:

1) производные Д-операторы используют (могут использовать) общие входные данные D_A;

2) выходные данные исходного Д-оператора D'_A – результат последовательного выполнения результирующих Д-операторов, т. е. они оба продуцируют (могут продуцировать) эти данные;

3) второй Д-оператор продолжает обработку данных (или некоторого их подмножества), начатую первым Д-оператором;

4) для обработки данных в подавляющем большинстве алгоритмов используются вспомогательные данные, которые, не являясь ни входными, ни выходными, служат для преобразования первых во вторые.

Для того чтобы учесть приведенные аспекты декомпозиции Д-операторов, запишем выражение из аксиомы в соответствии со структурой данных, предложенной в определении 3, и определим свойства обрабатываемых данных, исходя из их свойств, сформулированных в упомянутых определении и аксиоме.

Определение 7. Данные в выражении вида

$$(\overset{\frown}{D}_A,\hat{D}_A)A(\hat{D}'_A\tilde{D}'_A)=$$
$$=(\overset{\frown}{D}_B,\hat{D}_B)B(\hat{D}'_B,\tilde{D}'_B)*(\overset{\frown}{D}_C,\hat{D}_C)C(\hat{D}'_C,\tilde{D}'_C)$$

обладают следующими свойствами:
$\overset{\frown}{D}_A=\overset{\frown}{D}_B\cup\overset{\frown}{D}_C$, $\hat{D}_B\subseteq\hat{D}_A$, $\hat{D}_C\cap\hat{D}_A\neq\varnothing$, $\hat{D}'_B\cap\hat{D}'_A\neq\varnothing$, $\hat{D}'_C\subseteq\hat{D}'_A$, $\tilde{D}'_B\cap\tilde{D}'_A\neq\varnothing$, $\tilde{D}'_C\subseteq\tilde{D}'_A$.

Теперь рассмотрим последовательно перечисленные аспекты декомпозиции Д-операторов и определим состав и свойства данных, которые позволят реализовать операцию декомпозиции с учетом перечисленных аспектов.

Для того чтобы учесть первый аспект декомпозиции, определим (детализуем) состав и свойства входных данных Д-операторов с учетом состава и свойств данных, введенных в определении 7.

Определение 8. Входные данные Д-оператора $(\overset{\frown}{D}_A,\hat{D}_A)A(\hat{D}'_A\tilde{D}'_A)$ представляют собой объединение следующих подмножеств:

$$\overset{\frown}{D}_A={}_A\overset{\frown}{D}_B\cup{}_A\overset{\frown}{D}_{B,C}\cup{}_A\overset{\frown}{D}_C,$$
$$\hat{D}_A={}_A\hat{D}_B\cup{}_A\hat{D}_{B,C}\cup{}_A\hat{D}_C,$$

где

$${}_A\overset{\frown}{D}_B\subseteq\overset{\frown}{D}_B,\ {}_A\hat{D}_B\subseteq\hat{D}_B,\ {}_A\overset{\frown}{D}_C\subseteq\overset{\frown}{D}_C,$$
$${}_A\hat{D}_C\subseteq\hat{D}_C,\ \overset{\frown}{D}_B\supseteq{}_A\overset{\frown}{D}_{B,C}\subseteq\overset{\frown}{D}_C,$$
$$\hat{D}_B\supseteq{}_A\hat{D}_{B,C}\subseteq\hat{D}_C.$$

Из определения следует, что

$$\overset{\frown}{D}_B={}_A\overset{\frown}{D}_B\cup{}_A\overset{\frown}{D}_{B,C},\ \hat{D}_B={}_A\hat{D}_B\cup{}_A\hat{D}_{B,C},$$
$$\overset{\frown}{D}_C={}_A\overset{\frown}{D}_{B,C}\cup{}_A\overset{\frown}{D}_C,\ \hat{D}_C={}_A\hat{D}_{B,C}\cup{}_A\hat{D}_C, \qquad (7)$$

а данные ${}_A\overset{\frown}{D}_C$ и ${}_A\hat{D}_C$ поступают на вход Д-оператора $(D_C)C(D'_C)$, минуя Д-оператор $(D_B)B(D'_B)$. Принадлежность данных к некоторому множеству входных данных в определении отражена соответствующими индексами.

Таким образом, в общем случае входные данные исходного Д-оператора как распределяются между производными Д-операторами, так и присутствуют на входе обоих Д-операторов. Поскольку любое из подмножеств, образующих множество входных данных, может быть пустым, то очевидно, что входные данные исходного Д-оператора, как исходные $\widehat{D}_A$, так и проходные $\hat{D}_A$, могут использоваться как двумя производными Д-операторами, так и любым одним из них.

Заметим, что множества проходных данных ${}_A\hat{D}_B$, ${}_A\hat{D}_{B,C}$, ${}_A\hat{D}_C$, на входе Д-операторов, в соответствии с определением 3, присутствуют и на их выходе. Учитывая это, проходные данные ${}_A\hat{D}_B$ на выходе Д-оператора $(D_B)B(D'_B)$ обозначим ${}_B\hat{D}'_A$, а проходные ${}_A\hat{D}_C$ данные на выходе Д-оператора $(D_C)C(D'_C)$ $-{}_C\hat{D}'_A$. Обозначение для проходных данных ${}_A\hat{D}_{B,C}$ будет введено далее.

Для того чтобы учесть второй аспект декомпозиции, определим состав и свойства данных на выходе Д-оператора $(D_B)B(D'_B)$.

Определение 9. На выходе Д-оператора $(D_B)B(D'_B)$ из множества производных данных выделит подмножество ${}_B\tilde{D}'_A \supseteq \tilde{D}'_B$ такое, что ${}_B\tilde{D}'_A \subseteq \tilde{D}'_A$ и ${}_B\tilde{D}'_A \cap D_C = \varnothing$. Множество проходных данных ${}_B\hat{D}'_A$, такое, что ${}_B\hat{D}'_A \subseteq \hat{D}'_A$ и ${}_B\hat{D}'_A \cap D_C = \varnothing$.

То есть, определенные данные поступают с выхода Д-оператора $(D_B)B(D'_B)$ на выход исходного Д-оператора, минуя Д-оператор $(D_C)C(D'_C)$.

Поскольку $D'_C \neq \varnothing$, так как в противном случае Д-оператор $(D_C)C(D'_C)$ будет неопределенным (см. определение 2), то из определения 9 следует, что выходные данные Д-оператора $(D_A)A(D'_A)$ состоят из выходных данных D'_C Д-оператора $(D_C)C(D'_C)$, которые дополняются некоторыми подмножествами выходных данных Д-оператора $(D_B)B(D'_B)$, т. е.

$$D'_A = D'_C \cup {}_B\hat{D}'_A \cup {}_B\tilde{D}'_A. \qquad (8)$$

Для того чтобы учесть третий аспект декомпозиции введем понятие информационной связи (в дальнейшем связи) между Д-операторами.

Определение 10. Д-операторы $(D_B)B(D'_B)*(D_C)C(D'_C)$ связаны, если для них выполняется соотношение: $D'_B \cap D_C \neq \varnothing$.

Таким образом, в организации связи участвует некоторое подмножество выходных данных Д-оператора $(D_B)B(D'_B)$, которое поступает на вход Д-оператора $(D_C)C(D'_C)$, что позволяет ему продолжить их обработку.

При этом будем говорить, что данные передаются с выхода предшествующего Д-оператора на вход следующего. И будем полагать, что данные изменяются только в результате выполнения Д-операторов, а в процессе передачи остаются неизменными. То есть, данные на выходе Д-оператора идентичны одноименным данным на входе следующего Д-оператора.

Теперь определим, в соответствии с определениями 10 состав и свойства данных, связывающих Д-операторы $(D_B)B(D'_B)$ и $(D_C)C(D'_C)$. При этом будем учитывать, что в качестве таких данных могут выступать как проходные, так и производные данные.

Определение 11. Множество проходных данных ${}_A\hat{D}_{B,C}$, которое на выходе и входе Д-операторов $(D_B)B(D'_B)$ и $(D_C)C(D'_C)$ обозначим ${}_A\hat{D}'_{B,C}$, а на выходе последнего ${}_A\hat{D}''_{B,C}$ такое, что для него выполняется соотношение

$$({}_A\hat{D}_{B,C} \subseteq \hat{D}_B) = ({}_A\hat{D}'_{B,C} \subseteq \hat{D}'_B) =$$
$$= ({}_A\hat{D}_{B,C} \subseteq \hat{D}_C) = ({}_A\hat{D}''_{B,C} \subseteq \hat{D}'_C),$$

т. е., данные, образующие это множество, играют роль связывающих, а множества ${}_A\hat{D}_{B,C}, {}_A\hat{D}'_{B,C}, {}_A\hat{D}''_{B,C}$, являются одним множеством, по-разному обозначенным на входах и выходах Д-операторов. В общем случае эти множества неидентичны, т. е. ${}_A\hat{D}_{B,C} >< {}_A\hat{D}'_{B,C} >< {}_A\hat{D}''_{B,C}$. На выходе

Д-оператора $(D_B)B(D'_B)$ выделим подмножество производных данных ${}_B\hat{\tilde{D}}_C$ такое, что $\tilde{D}'_B \supseteq {}_B\hat{\tilde{D}}_C = {}_B\hat{\tilde{D}}_C \subseteq \hat{D}_C$, которое на выходе Д-оператора $(D_C)C(D'_C)$ обозначим ${}_B\hat{\tilde{D}}'_C$, и которое обладает следующими свойствами:

$$ {}_B\hat{\tilde{D}}_C \cap {}_B\tilde{D}'_A = \varnothing,\ {}_B\hat{\tilde{D}}_C \cap {}_A\hat{D}'_{B,C} = \varnothing, $$

$$ {}_B\hat{\tilde{D}}_C \cap {}_A\hat{D}'_C = \varnothing. $$

Таким образом, данные, полученные на выходе Д-оператора $(D_B)B(D'_B)$, могут быть подвержены дальнейшей обработке, т. е., обработка данных множества ${}_A\hat{D}_{B,C}$ начатая Д-оператором $(D_B)B(D'_B)$ и множества данных ${}_B\hat{\tilde{D}}_C$, продуцируемого этим Д-оператором, продолжается Д-оператором $(D_C)C(D'_C)$.

Исходя из определений 9 и 11, запишем состав выходных данных Д-оператора $(D_C)C(D'_C)$ в виде

$$ D'_C = {}_A\hat{D}'_C \cup {}_B\hat{\tilde{D}}'_C \cup {}_A\hat{D}''_{B,C} \cup \tilde{D}'_C, \qquad (9) $$

а исходя из соотношений (8) и (9) – состав выходных данных Д-оператора $(D_A)A(D'_A)$ в виде

$$ D'_A = {}_B\hat{D}'_A \cup {}_B\tilde{D}'_A \cup {}_C\hat{D}'_A \cup {}_B\hat{\tilde{D}}'_C \cup {}_A\hat{D}''_{B,C} \cup \tilde{D}'_C. \qquad (10) $$

Для реализации четвертого аспекта декомпозиции, нам необходимы связывающие данные ${}_B\breve{D}_C$ такие, которые не являлись бы входными и выходными данными исходного Д-оператора $(D_A)A(D'_A)$. Однако, будем утверждать, что такие данные не могут быть построены.

Утверждение. Связывающие данные ${}_B\breve{D}_C \subseteq \tilde{D}'_B$ такие, что ${}_B\breve{D}_C \cap D_C \neq \varnothing$ и ${}_B\breve{D}_C \cap D_A = \varnothing$, ${}_B\breve{D}_C \cap D'_A = \varnothing$, не могут быть введены в рамках предлагаемых формализмов, так как требуемое сочетание свойств нереализуемо.

Доказательство. Множество входных данных Д-оператора $(D_C)C(D'_C)$, в соответствии с определениями 3, представляет собой объединение подмножеств $D_C = \hat{D}_C \cup \hat{D}_C$, обладающих, в соответствии с определением 7, следующими свойствами $\hat{D}_C \subseteq \hat{D}_A$, $\hat{D}_C = \hat{D}'_C \subseteq \hat{D}'_A$. Таким образом, если ${}_B\breve{D}_C \cap \hat{D}_C \neq \varnothing$, то не выполняется соотношение ${}_B\breve{D}_C \cap D_A = \varnothing$, а если ${}_B\breve{D}_C \cap \hat{D}_C \neq \varnothing$, то не выполняется соотношение ${}_B\breve{D}_C \cap D'_A = \varnothing$.

Возникшее противоречие вызвано тем, что необходимые нам данные локализованы в исходном Д-операторе $(D_A)A(D'_A)$ и не специфицированы в качестве его входных и выходных данных. Такой тип данных в предшествующих рассуждениях не рассматривался.

Следствие. Для того чтобы разрешить возникшее противоречие, необходимо набор используемых данных дополнить данными нового типа - промежуточными.

Введем новый тип данных следующим образом.

Определение 12. Множество входных данных Д-оператора $(D_C)C(D'_C)$ запишем в виде $D_C = \hat{D}_C \cup \hat{D}_C \cup {}_B\breve{D}_C$, т. е. дополним его множеством ${}_B\breve{D}_C$, а из множества производных данных на выходе Д-оператора $(D_B)B(D'_B)$ выделим подмножество связывающих данных ${}_B\breve{D}_C$ такое, что $\tilde{D}'_B \supseteq {}_B\breve{D}_C = {}_B\breve{D}_C \subseteq D_C$. Множество данных ${}_B\breve{D}_C$, которые назовем промежуточными, обладает следующими свойствами:

$$ {}_B\breve{D}_C \cap \hat{D}_C = \varnothing,\ {}_B\breve{D}_C \cap \hat{D}_C = \varnothing, $$

$$ {}_B\breve{D}_C \cap D'_C = \varnothing,\ {}_B\breve{D}_C \cap {}_B\tilde{D}'_A = \varnothing, $$

$$ {}_B\breve{D}_C \cap {}_B\hat{\tilde{D}}_C = \varnothing. $$

Теперь, исходя из определения 12, докажем, что построенные данные обладают свойствами необходимыми для промежуточных данных.

Теорема 2. Для промежуточных данных $\breve{D}_C$ выполняются соотношения ${}_B\breve{D}_C \cap D_A = \varnothing$ и ${}_B\breve{D}_C \cap D'_A = \varnothing$, то есть они не являются входными и выходными данными исходного Д-оператора.

Доказательство.

Чтобы доказать, что промежуточные данные ${}_B\breve{D}_C$ не пересекаются с выходными данными D'_A Д-оператора $(D_A)A(D'_A)$ рассмотрим состав выходных данных $D'_A = D'_C \cup {}_B\hat{D}'_A \cup {}_B\tilde{D}'_A$, приведенный в (8). Из определения 12 следует, что ${}_B\breve{D}_C \cap D'_C = \varnothing$ и ${}_B\breve{D}_C \cap {}_B\tilde{D}'_A = \varnothing$, а из определения 9, что ${}_B\hat{D}'_A \cap D_C = \varnothing$. Но ${}_B\breve{D}_C \subseteq D_C$ и, таким образом, ${}_B\breve{D}_C \cap {}_B\hat{D}'_A = \varnothing$. Поскольку множество ${}_B\breve{D}_C$ не пересекается, ни с одним подмножеством, образующим множество D'_A, то соотношение ${}_B\breve{D}_C \cap D'_A = \varnothing$ выполняется.

Чтобы доказать, что промежуточные данные ${}_B\breve{D}_C$ не пересекаются с входными данными D'_A Д-оператора $(D_A)A(D'_A)$ рассмотрим состав этих данных, приведенный в определении 8,

$$\overset{\frown}{D}_A = {}_A\overset{\frown}{D}_B \cup {}_A\overset{\frown}{D}_{B,C} \cup {}_A\overset{\frown}{D}_C ,$$

$$\hat{D}_A = {}_A\hat{D}_B \cup {}_A\hat{D}_{B,C} \cup {}_A\hat{D}_C .$$

Из определения 12 известно, что ${}_B\breve{D}_C \cap \overset{\frown}{D}_C = \varnothing$ откуда, с учетом (7) следует, что ${}_B\breve{D}_C \cap {}_A\overset{\frown}{D}_C = \varnothing$ и ${}_B\breve{D}_C \cap {}_A\overset{\frown}{D}_{B,C} = \varnothing$. Известно также, что ${}_B\breve{D}_C \subseteq \tilde{D}'_B$, но $\tilde{D}'_B \cap D_B = \varnothing$ (см. определение 3) откуда, с учетом (7) следует, что ${}_B\breve{D}_C \cap {}_A\overset{\frown}{D}_B = \varnothing$ и, таким образом, ${}_B\breve{D}_C \cap \overset{\frown}{D}_A = \varnothing$.

Из соотношения $\tilde{D}'_B \cap D_B = \varnothing$ также следует, что $\tilde{D}'_B \cap \hat{D}_B = \varnothing$ и, таким образом, учитывая (7) ${}_B\breve{D}_C \cap {}_A\hat{D}_B = \varnothing$ и ${}_B\breve{D}_C \cap {}_A\hat{D}_{B,C} = \varnothing$. Так как ${}_B\breve{D}_C \cap \hat{D}_C = \varnothing$ (см. определение 12), то ${}_B\breve{D}_C \cap {}_A\hat{D}_C = \varnothing$ и, таким образом, ${}_B\breve{D}_C \cap \hat{D}_A = \varnothing$. Из соотношения $D_A = \overset{\frown}{D}_A \cup \hat{D}_A$ (см. определение 3) следует, что соотношение ${}_B\breve{D}_C \cap D_A = \varnothing$ выполняется.

Теорема доказана.

В результате выполненных построений, можем записать выражение из определения 7 в виде

$$({}_A\overset{\frown}{D}_B, {}_A\overset{\frown}{D}_C, {}_A\overset{\frown}{D}_{B,C}, {}_A\hat{D}_B, {}_A\hat{D}_C, {}_A\hat{D}_{B,C}) A({}_B\tilde{D}'_A,$$

$${}_B\hat{D}'_A, {}_C\hat{D}'_A, {}_A\hat{D}''_{B,C}, {}_B\hat{\tilde{D}}'_C, \tilde{D}'_C) =$$

$$=({}_A\overset{\frown}{D}_B, {}_A\overset{\frown}{D}_{B,C}, {}_A\hat{D}_B, {}_A\hat{D}_{B,C}) B({}_B\tilde{D}'_A,$$

$${}_B\hat{D}'_A, {}_B\breve{D}_C, {}_B\hat{\tilde{D}}_C, {}_A\hat{D}'_{B,C}) * ({}_A\overset{\frown}{D}_C, {}_A\overset{\frown}{D}_{B,C},$$

$${}_A\hat{D}_C, {}_A\hat{D}'_{B,C}, {}_B\hat{\tilde{D}}_C, {}_B\breve{D}_C) C({}_C\hat{D}'_A, {}_A\hat{D}''_{B,C}, {}_B\hat{\tilde{D}}'_C, \tilde{D}'_C),$$

т. е. в виде, когда в декомпозиции Д-операторов специфицированы все типы данных, необходимые для реализации этой операции.

Заключение

В данной работе предложен подход к формализации данных и построена такая структура данных, которая позволяет учесть все перечисленные аспекты декомпозиции операторов. При этом получены и доказаны некоторые свойства формализованных данных, которые могут быть использованы для контроля корректности операции декомпозиции.

Полученные результаты позволяют обратиться к решению следующих задач: доказательство необходимых условий корректности реализации операции декомпозиции; согласование процессов декомпозиции Д-операторов и детализации данных; исследование возможностей преобразования алгоритмов.

Решение этих задач в рамках предлагаемого формального аппарата и является направлением дальнейших исследований.

1. *Турский В.* Методология программирования. – М.: Мир, 1981. – 264 с.
2. *Ющенко Е.Л., Цейтлин Г.Е., Грицай В.П., Терзян Т.К.* Многоуровневое структурное проектирование программ: Теоретические основы, инструментарий. – М.: Финансы и статистика, 1989. – 208 с.
3. *Глушков В.М.* Теория автоматов и формальные преобразования микропрограмм // Кибернетика, 1965. – № 5. – С. 1 – 10.
4. *Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л.* Алгебра. Языки. Программирование. – К.: Наук. думка, 1978. – 319 с.

5. *Акуловский В.Г.* Расширенная алгебра алгоритмов // Проблеми програмування. – 2007. – № 3. – С. 3 – 15.
6. *Акуловский В.Г.* Формализация взаимосвязей операторов и данных в рамках расширенной алгебры алгоритмов // Кибернетика и системный анализ. – 2008. – № 6. – С. 170 – 182.
7. *Акуловский В.Г.* Некоторые подходы к контролю и преобразованию алгоритмов на основе анализа специфицируемых данных // Проблеми програмування. – 2008. – № 4. – С. 84 – 93.
8. *Акуловский В.Г.* Некоторые аспекты формализации архитектурного этапа разработки алгоритмов // Проблеми програмування. – 2009. – № 2. – С. 3 – 11.

Получено 12.10.2009

Об авторе:

Акуловский Валерий Григорьевич,
кандидат технических наук,
доцент кафедры информационных систем и технологий.
e-mail: valeryakulovskiy@rambler.ru

Место работы автора:

Академия таможенной службы Украины.
49000, Днепропетровск, ул. Дзержинского 2\4. Тел/факс. канцелярия – (0562) 45 5596
e-mail :academy@amsu.dp.ua