УДК 004.424, 004.415

В.Г. Колесник

DS-ТЕОРИЯ. ИССЛЕДОВАНИЕ ФАКТОРОВ ДЕЛЕНИЯ Р-ДАННЫХ ДЛЯ ГЕНЕРАЦИИ ПРИКЛАДНЫХ АЛГОРИТМОВ. ЧАСТЬ 1

Описаны виды деления Р-данных и рассмотрены факторы размещения их фрагментов и компонент. Для всех вариантов деления Р-данных описаны изменения в канонический алгоритм необходимые для их объединения. Изменения в алгоритме в плане сложности – это и примитивы в несколько повелительных операторов, и алгоритмические конструкции с циклами и управлением. Для построения алгоритмических конструкций предложен механизм синтеза – привязка по уровням дерева алгоритма. Для сравнительного анализа зависимости между схемой декомпозиции и прикладным алгоритмом предложено понятие АКУ-обусловленности как более подходящее, чем изоморфизм графов. Показано, что описание вариантов и факторов деления Р-данных имеет декларативный характер.

Введение

В работе [1] изложена идея, заключающаяся в том, что вместо попыток исследования множества реальных прикладных алгоритмов и программ следует исследовать множество схем декомпозиции (DS). Последнее множество значительно меньше множества реальных алгоритмов и программ. Также DS проще соответствующего алгоритма записанного с помощью алгоритмического языка. Соответственно, вместо проектирования алгоритмов и написания программ предпочтительнее проектировать DS. Но для этого необходимо иметь механизм преобразования DS в реальную программу. В работе [2] продемонстрирована возможность подобного преобразования. Канонический алгоритм, построенный на основании дерева полной схемы декомпозиции (DPS), задает структуру программы в целом. DS имеет в качестве компонент (строительных клеток) алгоритмические конструкции узла (АКУ). АКУ преобразуется в текст программы. Одна АКУ реализуется одним параграфом. Вся DS превращается в логически завершенную программу. В работе [2] показано, как размещение Р-данных на А-ленте влияет на содержимое параграфа.

В этой работе рассматривается еще один аспект работы с Р-данными. А-данные и С-данные, являющиеся аргументами при реализации А-зависимостей, перед началом расчета программы ЭВМ могут быть разделены и размещены в различных подобластях. В практике электронной обработки данных это значит, что группы и совокупности данных, могут быть размещены в различных файлах. Файлы же могут быть расположены как на одном носителе, так и на нескольких носителях. Данные могут быть расположены в различных базах данных. Кроме того, вместо файлов могут быть потоки данных [3], поступающих с различных источников – от датчиков, с интернета и т. п. Соответственно, Р-данные, как абстракция реальных данных, тоже могут быть расположены на различных носителях. В DS-теории в качестве носителей областей и А-фрагментов, содержащих фрагменты Р-данных, рассматривается абстрактная А-лента. Размещение Р-данных на А-ленте отличается от размещения реальных данных на реальных носителях. По сравнению с алгоритмами доступа к Р-данным на А-ленте, существуют различные факторы, которые следует учитывать при проектировании реальных алгоритмов для доступа к реальным данным. Факторы, которые учитываются при размещении разделенных Р-данных в областях или (и) на различных носителях, называются алгоритмически релевантными факторами деления Р-данных (АРФД).

Цель данной работы – показать обобщенную картину деления Р-данных, а также описать (очертить) группу алгоритмов объединения разделенных Р-данных –

алгоритмы слияния Р-данных. Описать группу АРФД и то, как эти факторы влияют на канонический алгоритм[1]. Описать те изменения, которые необходимо внести в канонический алгоритм, чтобы собрать Р-данные для реализации расчета А-зависимостей в виде одного непрерывного потока, так как будто они поступают с одной А-ленты.

Изменения, которые должны быть внесены в канонический алгоритм, это операторы или АК. Далее они будут либо полностью приведены, либо бегло описаны. Технически в условиях статьи невозможно показать весь набор изменений обусловленных группой АРФД. При анализе алгоритма объединения Р-данных, когда есть два или более разделенных Р-данных, будет обращаться внимание на то, появляется или нет синергетический эффект. То есть, не увеличивается ли непропорционально сложность из-за включения двух или более АК для объединения разделенных Р-данных. Синергетический эффект – это дополнительное препятствие для процедур синтеза.

1. Простейшие ситуации деления Р-данных и алгоритмы их объединения

Разделены могут быть А-данные и С-данные всех видов – простые, расширенные и сложные. Ситуации размещения фрагментов А-данных в областях на А-лентах возможны такие же, как и для А-данных, которые не разделены. А именно: все виды порядка размещения и все виды совместного размещения [2]. Аналогично ситуации размещения фрагментов С-данных в областях на А-лентах возможны такие же как и для неразделенных С-данных. В областях размещения фрагментов А-данных и С-данных наряду с последними могут быть Р-данные, которые являются непродуктивными для исходной DS. Фрагменты разделенных Р-данных в областях размещения являются просто А-фрагментами. А-фрагменты могут быть как условными, так и обязательными. Часть Р-данного может быть размещена не одной А-ленте, а другая на другой А-ленте. Первая А-лента называется основной, а вторая – смежной. Смежных А-лент может быть более одной[2].

Р-данные могут быть разделены перед их обработкой, но может быть необходимость их разделить после обработки. В работе рассматривает только один аспект – объединение Р-данных перед обработкой.

1. **Простое А-данное.** Тот факт, что простое А-данное разделено, обозначает следующее. А-данное (А) исходного дерева типов свойств (FS) состоит, по крайней мере, из двух Эл-данных: A = {B,C} (рис. 1, а). Одно из этих Эл-данных (В) размещено в записи (R_1) на одной из А-лент, а другое из Эл-данных (С) размещено в записи (R_2) на другой А-ленте. Для того, чтобы соединить их для обработки, необходимо прочитать в А-память записи с каждой из А-лент. Эта процедура реализуется в процессе вычислений при условии, что вычислительное устройство, упоминаемое в [1] в состоянии читать информацию с более чем одной А-ленты. Текст соответствующего фрагмента программного текста содержит два оператора READ[3]. В пределах этой статьи факт деления А-данного на чертеже изображается стрелкой, направленной снизу вверх, указывающей на границу разделения А-данного.

Если Эл-данное в записи на смежной А-ленте только одно из нескольких и имя не совпадает с именем в исходном А-данном, то необходимо уточнять, какое именно Эл-данное является частью разделенного исходного А-данного. В примере (рис. 1, б) – Эл-данное М в записи (R_1) на первой А-ленте – это Эл-данное В исходного А-данного А, а Эл-данное Z в записи (R_2) на второй А-ленте – это Эл-данное С в составе А-данного А. Если на одной из

[1] В истории математики есть прецедент подобного подхода в исследованиях. В какой-то момент стало понятно, что непродуктивно искать универсальные решения алгебраических уравнений пятой, шестой и больших степеней. Вместо этого были предприняты попытки исследовать свойства коэффициентов, от которых зависят решения. Это дало возможность ответить на вопрос о возможности нахождения корней в радикалах в принципе.

[2] В большинстве случаев между основной и смежной А-лентам может отсутствовать различие.

[3] Краткое описание алгоритмического языка в [2].

А-лент записей более одной, то искомую следует идентифицировать. Для этого надо указать ее тип. Для того, чтобы найти необходимую запись на смежной А-ленте необходимо построить алгоритмическую конструкцию (АК). Это цикл, в теле которого есть оператор чтения записи и условный оператор, который проверяет тип прочитанной записи. Цикл завершит работу, когда необходимая запись будет найдена. Если необходимые записи с компонентами исходного А-данного найдены на основной и смежных А-лентах, то дальше в тексте это факт будет звучать как “записи синхронизированы”.

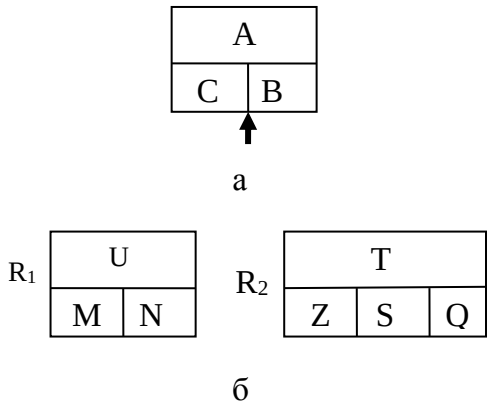

Рис. 1. Деление простого А-данного
FS: A = {C,B}
U={M,N}, T={Z,S,O}
RG_1: R_1 [U]
RG_2: R_2 [T]

Возможно, что А-данное содержит произвольное количество Эл-данных и разделено более одного раза. В результате деления получено несколько фрагментов А-данного и для размещения каждого фрагмента использована отдельная А-лента. То есть, для размещения фрагментов А-данного кроме основной использовано более одной смежной А-ленты. В таком случае синхронизация записей должна быть выполнена на всех А-лентах.

2. **Расширенное А-данное.** Тот факт, что расширенное А-данное разделено, обозначает следующее. А-данное А исходного дерева типов свойств состоит из произвольного количества Эл-данных, А-данных и простых или расширенных С-данных (рис. 2, а). Часть А-данного размещена в А-фрагменте (запись R_1, запись R_5 и подобласть R_2) на основной А-ленте, а другая часть размещена в А-фрагменте (запись R_3, R_6 и подобласть R_4) на смежной. АК в условиях примера на рис. 2 содержит оператор READ для чтения записи R_1, оператор READ для чтения первой записи из подобласти R_2, цикл для чтения остальных записей из R_2, оператор READ для чтения записи R_3 и оператор READ для чтения первой записи из подобласти R_4 и цикл для чтения остальных записей из R_4. Операторы READ и циклы соединены последовательным сочленением. Ниже приведен текст фрагмента на алгоритмическом языке (алг. 1.1).

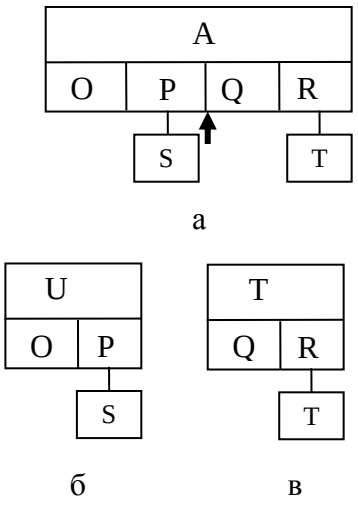

Рис. 2. Деление расширенного А-данного
FS: A={O,P,Q,R}, P = {∪S},
R = {∪T},
RG_1: R_1 [O], R_2 [P], R_5 [S]
RG_2: R_3 [Q], R_4 [R], R_6 [T]

Имена Р-данных в записях на А-лентах могут не совпадать с именами в FS. В таком случае необходимо установить соответствие между ними. В записях на А-лентах могут быть непродуктивные Р-данные. В этом случае никакие изменения в алгоритм не требуются. В подобластях на А-лентах могут быть непродуктивные записи, которые размещены вперемежку с искомыми. В таких случаях записи с искомыми Р-данными должны быть снабжены признаком, и при чтении записи без этого признака необходимо пропускать.

```
        AA: BEGIN
Чтение с первой А-ленты
        READ A1
        READ A1
        PERFORM BB (Конец подобласти
R2).
Чтение с второй А-ленты
        READ A2
        READ A2
        PERFORM CC (Конец подобласти
R4).
        END AA

        BB: BEGIN
        READ A1;
        END BB

        CC: BEGIN
        READ A2;
        END CC
```

Алгоритм 1.1

Возможна ситуация, когда искомая запись (R_1) и все записи, содержащие компоненты С-данного (например, Р) размещены в одном А-фрагменте, но на А-ленте есть и другие – непродуктивные А-фрагменты. В таком случае перед чтением искомого фрагмента, должна быть выполнена АК, которая обеспечит поиск этого А-фрагмента на смежной А-ленте. Успешное завершение поиска обозначается как “синхронизированы А-фрагменты”. При этом возможна ситуация, что внутри необходимого А-фрагмента есть непродуктивные записи. Существование таких записей влечет включение в текст фрагмента программы АК, которая позволит пропускать непродуктивные записи. Если искомые и непродуктивные А-фрагменты размещены в соответствии с некоторым порядком, то этот порядок в алгоритме может быть учтен, но его можно и не учитывать, но допустить, что искомый и непродуктивные А-фрагменты просто перемешаны.

Если расширенное А-данное разделено более чем на две части и для размещения разделенных частей используется кроме основной А-ленты более одной смежной А-ленты, то А-фрагменты должны быть синхронизированы на всех смежных А-лентах.

3. **Простое и расширенное С-данное.** Часть компонент (Q) С-данного (P) размещены в подобласти (U_1) на основной А-ленте (A1) (рис. 3). Эл-данное (Q) исходного С-данного размещено в записи (V_1). Другая часть компонент (Q) С-данного (P) размещены в подобласти (U_2) на смежной А-ленте (A2). Эл-данное (Q) исходного С-данного размещено в записи (V_2).

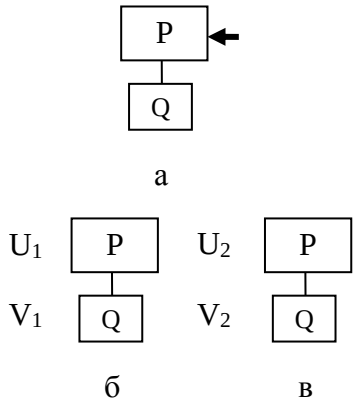

Рис. 3. Деление простого С-данного
FS: P = {∪Q}.
RG_1: U_1 [P], V_1 [Q]
RG_2: U_2 [P], V_2 [Q]

АК, которая обеспечит ввод искомых записей непрерывным потоком, в условиях примера на рис. 3 содержит оператор READ для чтения первой записи (V_1) с А-ленты (A1), цикл для чтения остальных записей(V_1) с той же подобласти, оператор READ для чтения первой записи (V_2) с А-ленты (A2) и цикл для чтения остальных записей (V_2) с той же подобласти. Операторы READ и циклы соединены последовательным сочленением.

Компонентами С-данного могут быть также простые А-свойства. Если А-свойства в подобластях на А-лентах и в FS совпадают по именам, то никакие изменения в алгоритме не нужны. Иначе для всех Эл-данных в FS и в записях, которые на А-лентах необходимо установить соответствие в именах.

В подобласти, как на основной, так и на смежной А-лентах могут быть записи, которые содержат непродуктивные Р-данные. А-фрагменты с продуктивными и непродуктивными Р-данными должны

иметь признаки, по которым их можно различать. Существование таких А-фрагментов влечет включение в текст фрагмента программы АК, которая позволит различать А-фрагменты с искомыми Р-данными. Если искомые и непродуктивные А-фрагменты размещены в соответствии с некоторым порядком, то этот порядок в алгоритме может быть учтен, но его можно и не учитывать, но допустить, что искомые и непродуктивные А-фрагменты просто перемешаны.

Возможна ситуация (рис. 4), когда а) для размещения разделенного расширенного С-данного использовано две А-ленты; б) в FS в А-данном, которое является компонентой расширенного С-данного, есть Эл-данное (Е), по возрастанию значения которого упорядочены записи; в) на основной (E_1) и смежной А-лентах (E_2) записи с искомым А-данным тоже упорядочены по значению этого же Эл-данного. Эл-данное, по значению которого упорядочены записи, называется ключом. Далее приведен алгоритм (алг. 1.2) для объединения упорядоченных по ключам записей с основной и смежной А-лент с сохранением порядка. Это так называемый алгоритм балансировки [4]. В условиях примера рис. 4. Эл-данное E_1 в записи V_1 соответствует Эл-данному Е в А-данном Т в FS, а в записи V_2 исходному Эл-данному Е соответствует Эл-данное E_2.

Параграф, соответствующий уровню исходного С-данного

```
AA: BEGIN
READ A1
READ A2
PERFORM  BB (У1+) AND (У2+).
END AA
```

Параграф, соответствующий уровню исходного А-данного.

```
BB: BEGIN
CASE
  IF (У2+) THEN
        IF (У1-) THEN READ A1
  IF (У2-) THEN
   CASE
    IF (У1+) THEN READ A2
    IF (У1-) THEN
        CASE
          IF (E2 > E1) THEN READ A1
          IF (E1 > E2) THEN READ A2
        ENDCASE
    ENDCASE
ENDCASE
END BB
```

Здесь:

У1+ – условие “достигнут конец подобласти R_1”;

У2+ – условие “достигнут конец подобласти R_2”;

У1- – условие “конец подобласти R_1 не достигнут”;

У2- – условие “конец подобласти R_2 не достигнут”;

Алгоритм 1.2

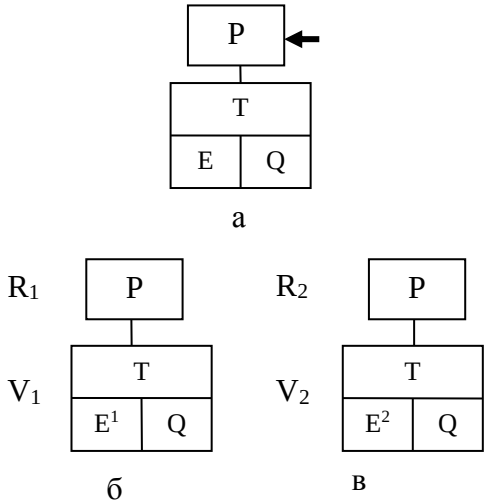

Рис. 4. Деление расширенного упорядоченного С-данного

FS: P = {∪T}, T={E,Q}

RG_1: R_1 [P], V_1 [T]

RG_2: R_2 [P], V_2 [T]

Если смежных А-лент более одной, то в параграф, который обеспечивает чтение записей непрерывным потоком без сохранения порядка, необходимо добавить АК для чтения с каждой дополнительной смежной А-ленты. Количество циклов в этих АК равно количеству смежных А-лент и одной основной. Если же необходимо объединить компоненты С-данного с сохранением порядка с нескольких А-лент, то в АЗП-параграфе по-

требуется АК, реализующая алгоритм балансировки для всех обрабатываемых А-лент. Размер и цели статьи не позволяют привести алгоритм для объединения С-данного разделенного и размещенного в записях на А-лентах, количество которых произвольно[4].

4. Расширенное А-данное как компонента С-данного. Расширенное А-данное – это компонента С-данного. Каждый экземпляр А-данного однообразно разделен на две части (рис. 5). Как первая часть А-данного, содержащая одни и те же Эл-данные, размещена в записи одного и того же типа (V_1) в подобласти (R_1) на основной А-ленте, так и вторая часть А-данного, содержащая остальные Эл-данные, размещена в записи одного и того же типа (V_2) в подобласти (R_2) на смежной А-ленте. Ниже приведен алгоритм (алг. 1.3) для объединения исходного А-данного (рис. 6, а).

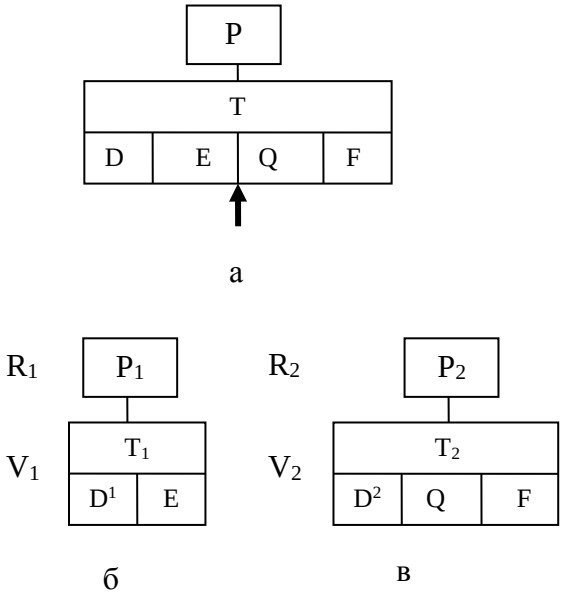

Рис. 5. Деление расширенного А-данного как компоненты С-данного
FS: P = {∪T}, T = {D,E,Q,F}
RG1: R_1 [P_1], V_1 [T_1]
RG2: R_2 [P_2], V_2 [T_2]

[4] В 80-е годы при программировании обработки линейных файлов или иерархических баз данных, если записи были разделены и размещены в нескольких файлах (более двух), а каждый файл на отдельной А-ленте, то написание алгоритма балансировки составляло определенную трудность и далеко не каждый программист мог его написать.

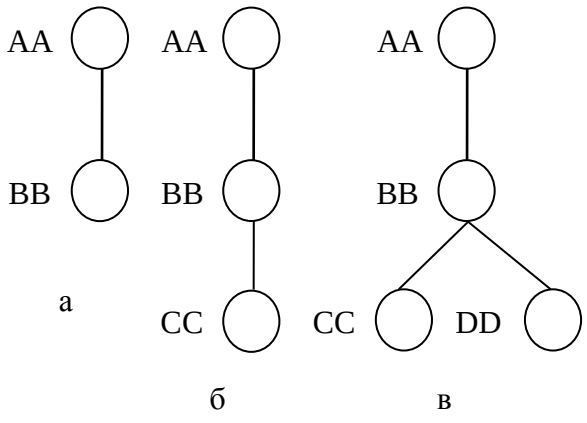

Рис. 6. Графы алгоритмов объединения разделенных А-данных

Параграф, соответствующий уровню исходного С-данного

```
AA: BEGIN
READ A1
READ A2
PERFORM  BB (У1+).
END AA
```

Параграф, соответствующий уровню исходного А-данного.

```
BB: BEGIN
    READ A1
    READ A2
END BB
```

Здесь:
У1+ – условие “достигнут конец подобласти R_1”.

Алгоритм 1.3

В этом алгоритме предполагается, что записи в областях на основной и смежной А-лентах одинаково упорядочены, и этот порядок будет сохранен при обработке. Между тем возможна ситуация, когда записи с фрагментами исходного А-данного на основной и исходной А-лентах имеют различный порядок. Алгоритм в этом случае будет построен исходя из того, в каком порядке будет необходимость их обрабатывать. Порядок может соответствовать порядку записей на основной или на смежной А-ленте. В любом из этих случаев в алгоритме необходимо добавить АК. Суть ее в следующем. Основной определяется А-лента, записи, на которой расположены в необходимом для обработки

порядке. В алгоритм добавляется цикл, который для каждой записи на основной А-ленте найдет соответствующую запись на смежной А-ленте. При этом поиск на смежной А-ленте каждый раз начинать с начала подобласти (рис. 6, б). Также в каждой из записей на основной и на смежной А-лентах должно Эл-данное (или А-данное), которое является ключом для их синхронизации.

Возможен вариант, когда порядок обработки записей не совпадает с порядком их размещения ни на основной, ни на смежной А-лентах. Для работы алгоритма необходимо обеспечить способ, с помощью которого будет указан порядок обработки записей. Возможно потребуется изменить исходную FS – добавить вспомогательный набор записей, определяющий порядок обработки. Алгоритм в таком случае будет следующим: после чтения записи со вспомогательного набора записей выполнить цикл для поиска необходимого фрагмента исходного А-данного на основной А-ленте и цикл для поиска необходимого второго фрагмента исходного А-данного на смежной А-ленте[5] (рис. 6, в). Особые виды порядка размещения записей с фрагментами А-данного в подобластях, как на основной, так и на смежной А-лентах, а именно, относительно начала или конца подобласти или относительно друг друга, не имеют значения, так как важно соответствие порядка на А-лентах.

Как на основной, так и на смежной А-лентах в подобластях, где расположены записи с фрагментами исходного А-данного, могут быть также и записи с непродуктивными Р-данными. В таком случае в алгоритм подобный алгоритму 1.3 необходимо внести изменения. Вместо оператора READ, который читает записи с А-ленты с непродуктивными Р-данными необходимо вставить АК, которая будет распознавать тип прочитанной записи и поставлять в поток для обработки только записи с фрагментом искомого А-данного.

Как на основной, так и на смежной А-лентах могут отсутствовать некоторые записи с фрагментами искомого А-данного. Такой вариант с фрагментами А-данного должен быть описан в FS и в описании размещения Р-данных в подобластях. Если в FS это не описано, то это значит, что FS – ошибочна. Если же описано, то следующим образом.

- Первый вариант. Некоторые записи отсутствуют только на одной из А-лент. Это значит, что исходное С-данное содержит как компоненты два типа А-данных – укомплектованное фрагментами и неукомплектованное фрагментами первой частью А-данного.
- Второй вариант. Исходное С-данное содержит как компоненты два типа А-данных – укомплектованное фрагментами и неукомплектованное фрагментами другой частью А-данного.
- Третий вариант. Исходное С-данное содержит как компоненты четыре типа исходного А-данного – укомплектованное фрагментами, неукомплектованное фрагментами одной части, неукомплектованное фрагментами другой части А-данного и неукомплектованное фрагментами обоих частей.

То, какой вариант предлагается для последующей обработки в потоке экземпляров исходного А-данного, определяется только после того, когда найдены (или не найдены) записи с фрагментами на обеих А-лентах.

При объединении записей с фрагментами исходного А-данного возможно сочетание вариантов: упорядочены или нет записи с фрагментами исходного А-данного, наличие записей непродуктивными Р-данными и отсутствие записей с фрагментами исходного А-данного. Это должно быть описано в FS и в описании размещения Р-данных в подобластях.

Исходное А-данное может быть разделено на фрагменты, которых больше чем два. В таком случае для размещения записей может быть использовано количество А-лент равное количеству типов фрагментов или меньшее этого количества. В последнем случае предполагается, что на одной А-ленте размещено более одного типа записи с фрагментами исходного А-данного. Каждый тип фрагмента сохра-

[5] В практике электронной обработки данных подобные задачи решаются с помощью сортировки или индексации данных.

няется в уникальном типе записи. Для всего этого разнообразия разделенного А-данного алгоритмы изменяются только механически. А именно, в одном и том же параграфе добавляются АК, которые описаны выше, кроме той ситуации, когда надо учитывать различающийся порядок размещения записей в подобластях. В том случае, когда надо учитывать различный порядок записей в подобластях на основной и (или) смежной А-лентах, появляется дополнительный цикл (циклы) и, соответственно, дополнительный параграф. В условиях примера (рис. 5 и алгоритм 1.3) – это параграф ВВ на рис. 6, б и в.

5. Алгоритмически-зависимый параграф. Как было показано в [2] каждая алгоритмическая конструкция узла (АКУ) реализуется как один параграф. Каждый параграф связан со “своей” АКУ или, точнее, каждый параграф зависит от “своей” АКУ. Также можно говорить, что каждый узел дерева алгоритма (DA) зависит от “своего” узла в DPS. Для того, чтобы подчеркнуть связь между конкретным АКУ (узлом DPS) и параграфом (узлом в DA), используется понятие “алгоритмически-зависимый” (АЗ). А именно, алгоритмически-зависимый параграф (АЗ-параграф), алгоритмически-зависимый узел (АЗ-узел), алгоритмически-зависимый узел-параграф (АЗ-узел-параграф). Это же понятие будет использоваться для иллюстрации связи между главным А-данным (узлом в FS) и параграфом (узлом в DA).

В следующей ситуации вводится еще одно определение связи между FS и алгоритмом. В главном А-данном есть некое С-данное. Главное А-данное соотносится с АЗ-параграфом. В С-данном есть компоненты, для обработки которых в алгоритме порождается параграф. Последний будет подчинен исходному АЗ-параграфу (связан с ним иерархическим сочленением). Этот подчиненный параграф по отношению к упомянутому С-данному будет называться алгоритмически-зависимым подчиненным (АЗП) параграфом, а именно, АЗП-параграф (АЗП-узел, АЗП-узел-параграф).

Если выполнение АЗ-параграфа инициируется другим параграфом, то последний называется алгоритмически зависимым управляющим параграфом (АЗУ-параграф) для исходного АЗ-параграфа. АЗУ-параграф и АЗ-параграф соединены иерархическим сочленением. Соответственно, существуют АЗУ-узел, АЗУ-узел-параграф. Для АЗ-узла (АЗ-параграфа) АЗУ-узел (АЗУ-параграф) может отсутствовать только в том случае, если АЗ-узел размещен на нулевом уровне AD. С точки зрения FS и DA как деревьев, АЗУ-узел это узел-родитель для АЗ-узла, а АЗП-узел – узел-потомок для того же АЗ-узла.

Если алгоритм, с помощью которого реализована некая DPS, содержит только АЗ-параграфы, количество которых совпадает с количеством узлов в DPS, то он называется АКУ обусловленной текстовой реализацией канонического алгоритма. Или в другой формулировке – алгоритм имеет свойство АКУ-обусловленности. Свойством АКУ-обусловленности обладает не каждый реальный алгоритм. Как правило в реальных алгоритмах появляются дополнительные вспомогательные параграфы.

2. Произвольное деление С-данных

1. **Деление С-данного в трехуровневом FS (первый вариант).** С-данное разделено и при этом оно является компонентой трехуровневого FS. С-данное, будучи в узле на нулевом уровне FS, имеет своими компонентами не простое А-свойство, а расширенное[6] (рис. 7, а). Это означает, что объединять надо не записи, размещенные в различных подобластях, а подобласти[7].

DA как и FS тоже имеет три уровня. Подобласти на различных А-лентах неупорядочены. В параграфе на нулевом уровне, который является АЗ-параграфом для узла, где разделено С-данное, органи-

[6] На рис. 7 только в пункте “а” детально изображается примерный состав FS. В остальных пунктах FS изображается сжато – только узлы, количество уровней FS, количество и места деления С-данных.

[7] Подобласти как и записи тоже могут быть упорядочены.

зованы циклы для объединения компонент С-данного. В параграфе первого уровня (рис. 7, в – параграфы BB и DD), вместо оператора READ, который читал бы очередную запись, организован цикл для чтения подобластей – компонентой разделенного С-данного. Содержимое параграфов первого и второго уровня (рис. 7, в – параграфы CC и EE) такое, как если бы С-данное на нулевом уровне FS не было бы разделено. Объединение разделенного С-данного для варианта неупорядоченных компонент реализовано только в АЗ-параграфе.

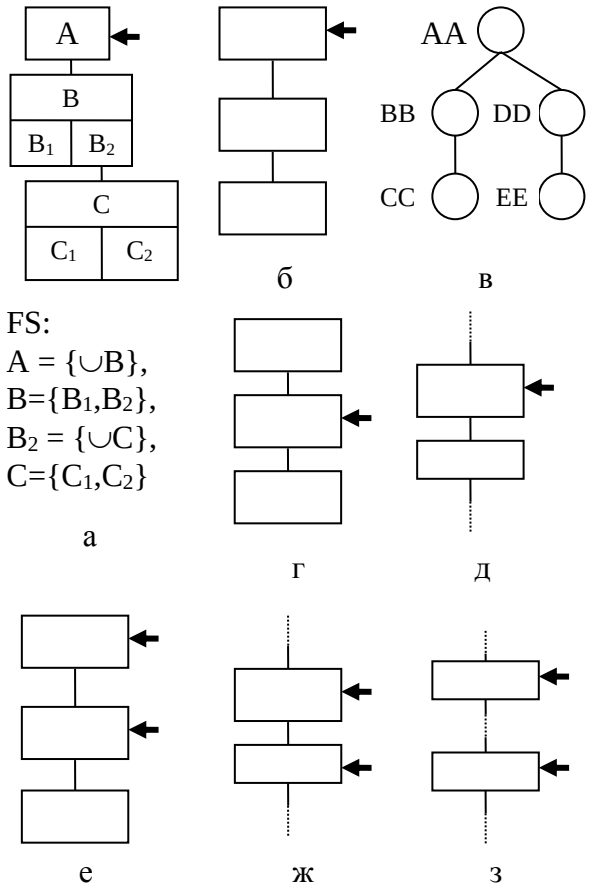

Рис. 7. Различные виды деления С-данных

Объединение разделенного С-данного для варианта упорядоченных компонент реализуется в АЗ-параграфе и в АЗП-параграфе. Деления С-данного не влияет на узел-параграф второго уровня.

2. **Деление С-данного в трехуровневом FS (второй вариант).** С-данное разделено и при этом оно является компонентой трехуровневого FS. В этом варианте разделенное С-данное размещено в узле на первом уровне и является компонентой С-данного (рис. 7, г). Компоненты С-данного в подобластях размещения неупорядочены и нет необходимости поставлять их для обработки в каком-либо порядке. В АЗ-параграфе организованы циклы для объединения компонент С-данного. В параграфе второго уровня, будут использованы операторы READ, для чтения очередных записей, содержащих компоненты С-данного. Содержимое параграфов нулевого и второго уровня такое, как если бы С-данное на первом уровне FS не разделено. Объединение разделенного С-данного для варианта неупорядоченных компонент реализовано только в параграфе на первом уровне алгоритма – в АЗ-параграфе для разделенного С-данного.

Объединение разделенного С-данного для варианта упорядоченных компонент реализуется в АЗ- и АЗП-параграфах. Факт деления С-данного для параграфа нулевого уровня (не в АЗ- или АЗП-параграфах) не влияет на алгоритм.

3. **Деление С-данного в многоуровневом FS.** С-данное разделено и при этом оно является компонентой многоуровневого FS (рис. 7, г). Количество уровней в FS произвольное (n уровней n > 3). С-данное может быть размещено в узлах начиная с нулевого уровня и заканчивая уровнем n - 2.

Если компоненты С-данного неупорядочены и нет необходимости поставлять их для обработки в определенном порядке, то в АЗ-параграфе будут два цикла, которые читают компоненты на основной и смежной А-лентах, а от АЗ-узла будут отходить две ветви. Циклы будут соединены последовательным сочленением. В DA от АЗ-узла будут отходить вниз две ветви, несмотря на то, что в FS от узла с разделенным С-данным отходит одна ветвь. Из-за этого DA не сохраняет АКУ-обусловленность.

Если компоненты С-данного упорядочены и их необходимо поставлять для обработки в определенном порядке, а разделенное С-данное размещено на уровне n - 2, то в АЗ-параграфе будут два цикла, которые читают компоненты на основной

и смежной А-лентах, а от АЗ-узла будет отходить вниз одна ветвь. В АЗП-параграфе АК, которые читают компоненты в определенном порядке, будут соединены альтернативным сочленением.

Если компоненты С-данного упорядочены и необходимо поставлять их для обработки в определенном порядке, а разделенное С-данное размещено на уровне n - 3 или выше, то в АЗ-параграфе будут два цикла, которые читают компоненты на основной и на смежной А-лентах, а от АЗ-узла будет отходить ветвь к АЗП-узлу. В АЗП-параграфе АК, которые читают компоненты в определенном порядке, будут соединены альтернативным сочленением. От АЗП-узла будут отходить вниз две ветви. Из-за этих двух ветвей (их более одной) и узлов на этих ветвях DA не сохраняет АКУ-обусловленность.

Независимо от того, на каком из возможных уровней FS будет расположено разделенное С-данное, и независимо от того упорядочены или нет компоненты С-данного на А-лентах (и надо ли их поставлять для обработки в определенном порядке), АК, что реализует объединение С-данного, будет расположена в АЗ-параграфе для узла FS на котором разделенное С-данное. Все параграфы расположенные выше в DA, будут такими, как будто разделения С-данного нет. Для упорядоченных компонент изменения в алгоритме, связанные с объединением С-данного, будут еще в АЗП-параграфе.

Если эти АЗП-узлы не конечные, то они будут началами ветвей. Различие между этими ветвями будет только в том, что в соответствующих АК операторы READ будут читать записи с различных А-лент.

4. **Деление двух С-данных в трехуровневом FS.** В трехуровневом FS одно С-данное разделено в узле на нулевом уровне, а другое в узле на первом уровне (рис. 7, е). АЗ-параграф нулевого уровня FS будет содержать АК, которая будет поставлять для обработки компоненты разделенного С-данного. Этими компонентами есть подобласти. АК будут соответствовать тому, упорядочены или нет разделенные компоненты на основной и смежной А-лентах. В АЗП-параграфе (АЗП-параграфах) используются АК для объединения С-данного, разделенного на первом уровне. Для АК нулевого уровня не имеет значения, какая именно использована АК в АЗП-параграфе – объединяет ли она поставляемую подобласть или только прочитывает ее с А-ленты. Обе АК, предназначенные для объединения разделенных С-данных, что размещены в АЗ-параграфе и в АЗП-параграфе независимы друг от друга. Их взаимное существование в этих параграфах не добавляет каких-либо дополнительных АК или операторов. То есть, существование более чем одной АК, предназначенных для объединения разделенных С-данных не порождает синергетического эффекта, – не добавляет сложности в алгоритм. Сложность растет только при описании RG. То есть, необходимо описать то, как размещены Р-данные на А-лентах, которых более двух.

Если С-данное разделено на нулевом уровне, то оно будет размещаться на двух А-лентах (Л1 и Л2). Далее, с точки зрения количества А-лент, при делении С-данного на первом уровне для окончательного размещения возможны два варианта. Первый. Р-данные размещены на трех А-лентах (Л1 и (Л2 и Л2а) или (Л1 и Л1а) и Л2. Второй. Р-данные размещены на четырех А-лентах ((Л1 и Л1а) и (Л2 и Л2а). Это должны быть различные А-ленты. Если же какие-то из А-лент будут совпадать (например, Л1 и Л2а или Л1 и Л2 или Л1а и Л2а), алгоритм будет многопроходным. При любом размещении в DA отсутствует АКУ-обусловленность.

5. Общий случай деления С-данных в многоуровневом FS. В многоуровневом FS (n уровней, n>3) два С-данных разделены в узлах на смежных уровнях (рис. 7, ж). Деление С-данного на уровне с меньшим номером может быть, начиная с нулевого и заканчивая уровнем, номер которого (n-2) на два меньше номера самого нижнего уровня. Деление С-данного на смежном с меньшим номером может быть, соответственно, начиная с первого и заканчивая уровнем, номер которого на один меньше номера самого нижнего уровня (n-1). При этом уровни с разделенными С-данными соединены вет-

вью. Независимо от того, на каком уровне разделено С-данное (от нулевого до n-2), для объединения разделенных компонент необходима одна из тех АК, что упомянуты выше, – для упорядоченных или неупорядоченных компонент. Никакие дополнительные операторы или АК не нужны. Также не нужны дополнительные АК или операторы в АЗП-параграфе. Ситуация с DA такая же как и в п. 4. АКУ-обусловленность не сохраняется.

В многоуровневом FS (n уровней, n>3) два С-данных разделены на уровнях с произвольными номерами. Разделенные С-данные могут быть на различных уровнях, которые не обязательно смежные (рис. 7, з). Деление может быть на уровнях, начиная с нулевого и заканчивая предпоследним (n-1). Как и в предыдущем случае для объединения разделенных компонент необходима одна из АК, – для упорядоченных или неупорядоченных компонент. Никакие дополнительные операторы или АК не нужны в обоих АЗ-параграфах. Ситуация с DA такая же как и в п. 4. АКУ-обусловленность не сохраняется.

С одним узлом FS может соотноситься более одного С-данного и некоторые из них разделены. Разделенных С-данных в этом узле более одного. То, как С-данные разделены, описано в RG. Для объединения компонент каждого из разделенных С-данных подбираются соответствующие АК. Содержимое каждой из этих АК независимо друг от друга. Синергетический эффект не порождается. АЗП-параграфы для объединения каждой группы компонент независимы друг от друга. АКУ-обусловленность не сохраняется.

FS содержит много ветвей. Каждая из ветвей содержит произвольное количество уровней. В узлах на любом из уровней, кроме узлов конечных по ветви, есть С-данные. Разделенные С-данные размещены в узлах, которые расположены по различным ветвям. АК и соответствующие АЗП-параграфы, необходимые для объединения компонент разделенных С-данных, независимы друг от друга и не порождают синергетический эффект. АКУ-обусловленность не сохраняется.

FS содержит произвольное количество узлов, на произвольном количестве уровней. Одно С-данное может быть разделено и размещено на более чем двух А-лентах. Если компоненты необходимо представить для обработки без сохранения порядка, то включаются АК, что содержат циклы для чтения компонент на соответствующих А-лентах. Количество АК равно количеству А-лент. Происходит механическое увеличение количества АК, соединенных последовательным сочленением. Если же компоненты необходимо представить для обработки в определенном порядке, то потребуется алгоритм балансировки. Алгоритм формируется независимо от того, есть какое-либо деление С-данных в узлах на других уровнях. В каждом из этих двух случаев не наблюдается синергетический эффект. АКУ-обусловленность не сохраняется.

1. *Колесник В.Г.* DS-теория как прототип теории прикладных алгоритмов // Проблеми програмування. – 2012. – № 1. – С. 17–33.
2. *Колесник В.Г.* DS-теория. Представление канонического алгоритма с помощью алгоритмического языка // Проблеми програмування. – 2015. – № 1. – С. 3–18.
3. *Jackson* structured programming [Электронный ресурс] // http://en.wikipedia.org/wiki/ Jackson_structured_programming.
4. *Сортировка* слиянием [Электронный ресурс] // https://ru.wikipedia.org/wiki/ Сортировка_слиянием.

Получено 20.04.2015

Об авторе:

Колесник Валерий Георгиевич,
старший научный сотрудник
кафедры АПП.

Место работы автора:

Донбасская государственная
машиностроительная академия.
г. Краматорск, ул. Шкадинова, 72, п/я 13.