



УДК 004.6

ПОЛЕ СТРУКТУР ДАННЫХ

© А. М. ВОЛОДИН, В. В. ДРОЖДИН

Пензенский государственный педагогический университет им. В. Г. Белинского,
кафедра прикладной математики и информатики
e-mail: a.m.volodin@gmail.com, drozhdin@yandex.ru

Володин А. М., Дрождин В. В. – Поле структур данных // Известия ПГПУ им. В. Г. Белинского. 2010. № 18 (22). С. 123–129. – Рассмотрены конструктивные элементы и уровни организации структур данных. На основе общей теории систем Ю.А. Урманцева разработана модель поля структур данных, максимально учитывающая разнообразие известных структур данных. Показано соответствие и полнота преобразований структур данных в поле структур данных преобразованиям объектов-систем в общей теории систем.

Ключевые слова: система, структура данных, поле структур данных, изомерия, элемент данных, параметр, признак, указатель.

Volodin A. M., Drozhdin V. V. – The field of data structures // *Izv. Penz. gos. pedagog. univ. im. V. G. Belinskogo*. 2010. № 18 (22). P. 123–129. – The constructive elements and organization levels of data structures are considered. Based on general systems theory written by Y.A. Urmantsev the model of data structures field that comprises the variety of known data structures is developed. The correspondence and completeness of data structures transformations in the field of data structures with objects-systems transformations in general systems theory is pointed out.

Keywords: system, data structure, field of data structures, isomerism, data element, parameter, characteristic, pointer.

Структура данных (СД) – это система, состоящая из элементов данных и отношений между ними, представляющих единое целое. В [1] установлено, что более сложные (нерегулярные) СД могут включать дополнительную информацию в виде параметров и признаков. Поэтому в общем случае СД могут строиться из **конструктивных элементов** следующих типов:

- элементы данных, способные содержать значения данных определенного размера;
- отношения (связи, указатели) между элементами, позволяющие получить доступ от одних элементов к другим элементам;
- параметры, позволяющие настраивать структуры на необходимое использование и обработку;
- признаки, информирующие одни элементы о свойствах других элементов.

Для спецификации СД будем использовать следующие обозначения: $D = \{d\}$ – множество элементов данных, $L = \{l\}$ – множество параметров, $C = \{c\}$ – множество признаков, $U = \{u\}$ – множество указателей.

Ниже приведено графическое изображение конструктивных элементов СД:



а) элемент данных б) параметр в) признак г) указатель.

Элементы данных могут содержать значения данных разных типов, в разной форме представления, разного размера и др.

Параметры задают количественные характеристики элементов и отношений между ними, например, для значений данных могут задаваться минимальное, максимальное или среднее значение, отклонение от среднего или интервал значений, для отношений между данными могут задаваться допустимое, текущее, ожидаемое или прогнозируемое (расчетное) количество элементов данных и др.

Признаки отражают специфические особенности отдельных конструктивных элементов и их групп. Пассивные признаки фиксируют (регистрируют) некоторые факты, которые слабо влияют на обработку данных, но впоследствии могут быть проанализированы и служить основой для модификации методов обработки данных. Активные признаки выражают некоторые правила, которые существенно влияют на эффективность обработки данных.

Отношения между конструктивными элементами задают их связность и могут реализовываться двумя способами: последовательным физическим размещением и ссылкой (физическим адресом) на требуемый элемент. Можно выделить одноуровневые указатели на предшествующий и последующий элементы, иерархические – на родителей и потомков (подчинен-

ные элементы), на выделенные элементы, обладающие определенными свойствами.

Приведенный набор конструктивных элементов и их свойств позволяют потенциально формировать огромное разнообразие как однородных, так и гетерогенных СД.

СД, содержащие только элементы данных и указатели, называются регулярными [1]. Все классические СД (множества, последовательности, деревья и др.) являются регулярными. Регулярные СД преимущественно ориентированы на равновероятный доступ к любому элементу данных.

В отличие от регулярных, квазирегулярные и нерегулярные [1] СД содержат также дополнительную (служебную или управляющую) информацию, представленную в виде параметров и признаков. Причинами возникновения нерегулярности СД являются неравномерность использования данных и специфика их обработки, фиксируемые в дополнительной информации.

В квазирегулярных СД дополнительная информация используется только в определенных ситуациях и несущественно влияет на логику обработки структуры. Примером квазирегулярных структур являются СД, используемые системой управления базами данных (СУБД) для хранения данных в таблицах и индексах, содержащие, например, информацию о количестве элементов и уровней в индексе.

В нерегулярных СД дополнительная информация не просто фиксирует особенности использования данных, а существенно влияет на их обработку, обеспечивая повышение ее надежности и эффективности. Нерегулярные СД строятся из гетерогенных (неоднородных) подструктур и элементов с разной организацией и дисциплиной обслуживания, которые могут быть по-разному упорядочены (частично упорядочены или неупорядочены), содержать или не содержать дубликаты и т.д.

СД целесообразно рассматривать как автономную систему, которая может возникать, формироваться, существовать в некотором неизменном виде или изменяться по определенным законам. Такие СД будут представлять собой реальные объекты.

Как любые другие реальные объекты, СД могут характеризоваться некоторым набором свойств, отражающих их структуру, организацию и поведение.

В теории динамических систем системы принято задавать с помощью набора переменных x_1, x_2, \dots, x_n , однозначно определяющих их состояния. Тогда множество возможных состояний системы называется ее **фазовым пространством** (ФП), а изменение системы представляет собой некоторую траекторию в ФП. Поведение динамической системы, задается дифференциальным уравнением $y = f[x(t)]$, где $x = (x_1, x_2, \dots, x_n)$ – n -мерный вектор фазовых координат, или вектор состояния системы, t – время, f – вектор-функция, которая при изменении времени t описывает фазовую траекторию изменения системы [2].

Фазовое пространство структуры данных (ФПСД) может быть задано с помощью фазовых ко-

ординат, представляющих ее различные состояния. В качестве фазовых координат могут выступать число элементов данных, тип и число связей между элементами, типы и количество признаков и параметров, методы доступа и обработки данных, а также другие логические и физические характеристики СД.

Однако учет в ФПСД только качественных и количественных характеристик СД и не позволяет однозначно задать все возможные СД. Например, с помощью ФПСД невозможно объяснить явление изомерии.

Две структуры данных S_1 и S_2 называются **изомерами** ($S_1 \cong S_2$), если они содержат одинаковый состав конструктивных элементов, но различные отношения между ними. Как следствие, структуры-изомеры будут иметь одни и те же фазовые координаты, но при этом будут отличаться друг от друга структурно.

К тому же, СД целесообразно рассматривать не как динамические, а как эволюционные системы, поведение которых не может быть описано аналитически.

Поэтому необходим системный подход к построению пространства состояний всех возможных СД, которое будем называть **полем структур данных** (ПСД).

Построение ПСД данных будем осуществлять на основе общей теории систем Ю.А. Урманцева (ОТС-У) [3, 4].

Основными понятиями ОТС-У являются понятия объекта-системы и системы объектов одного и того же i -того рода.

Объект-система (OS) – это композиция или единство элементов $m \subseteq M_{OS}$ построенная по отношениям (взаимодействиям) $r \subseteq R_{OS}$ и ограничивающим их условиям или законам композиции $z \subseteq Z_{OS}$. Тогда в общем виде OS представляется тройкой:

$$OS = \langle m, r, z \rangle.$$

Система объектов одного и того же i рода (R -система или RS) – это закономерное множество OS одного и того же качества (рода i), означающая, что каждая OS построена:

- из всех или части элементов $m \subseteq M^i$, выделенных из универсума U ($M^i \subseteq U$) по основаниям $a \subseteq A^i$;
- в соответствии со всеми или частью отношений $r \subseteq R^i$;
- в соответствии со всеми или частью законами композиции $z \subseteq Z^i$.

Тогда R -система представляется в виде:

$$RS^i = \{OS^i\} = \{ \langle m^i, r^i, z^i \rangle \mid m^i \subseteq M^i, r^i \subseteq R^i, z^i \subseteq Z^i \},$$

где $M^i = U \perp A^i$; \perp – операция выделения подмножества из множества.

Любой объект есть OS и каждая OS принадлежит хотя бы одной RS [3]. Поэтому $M_{OS} = \cup M^i$, $R_{OS} = \cup R^i$, $Z_{OS} = \cup Z^i$ R -систем, которым принадлежит OS .

Если OS принадлежит одной RS^i , строится на основе всех отношений R^i и удовлетворяет всем законам композиции Z^i , то модель OS будет иметь вид:

$$OS = \langle m, R^i, Z^i \rangle,$$

а в случае известной RS^i :

$$OS = \langle m \rangle.$$

Если OS принадлежит нескольким RS , но существенно проявляет свойства только некоторой RS^i , а принадлежность другим RS несущественно проявляется в OS ($M_{OS} \setminus M^i \rightarrow \emptyset$, $R_{OS} \setminus R^i \rightarrow \emptyset$, $Z_{OS} \setminus Z^i \rightarrow \emptyset$), то вследствие системного изоморфизма можно считать, что OS принадлежит только RS^i и отклонения в OS от RS^i составляют систему противоречий, которую будем называть системой исключений.

Таким образом, если OS принадлежит RS^i , то накопление исключений позволяет ей, при определенных условиях, перейти в RS^j , а если OS равноправно принадлежит нескольким RS , то она сочетает в себе их свойства.

Теперь можно построить R -систему на множестве всех RS^i . Для этого будем учитывать, что все RS^i базируются на:

- общем универсуме элементов U ;
- множестве возможных отношений единства между элементами R ;
- множестве возможных ограничений (законов композиции) Z .

Так как все RS^i строятся из универсума элементов U , отношений единства R и законов композиции Z , то все они имеют общую (универсальную) природу, а типы RS^i проявляются при их взаимодействии и в структуре OS . Поэтому можно определить отношение вхождения одной R -системы в другую.

Будем говорить, что RS^i полностью входит (включена) в RS^j и обозначать это как $RS^i \subseteq RS^j$, если $M^i \subseteq M^j$, $R^i \subseteq R^j$ и $Z^i \subseteq Z^j$, т.е. для формирования RS^j , включающей RS^i , достаточно удалить из Z^j или ослабить некоторые ограничения. Отношение включения по своей природе является отношением иерархии между R -системами.

RS^i частично входит (частично включена) в RS^j и обозначается как $RS^i \subset RS^j$, если $M^i \cap M^j \neq \emptyset$, $M^i \setminus M^j \neq \emptyset$, $R^i \cap R^j \neq \emptyset$, $R^i \setminus R^j \neq \emptyset$, $Z^i \cap Z^j \neq \emptyset$ и $Z^i \setminus Z^j \neq \emptyset$, т.е. для формирования RS^j , частично включающей RS^i , необходимо выделить в $OS^j \in RS^j$ части подсистем, соответствующие объектам-системам $OS^i \in RS^i$. Отношение частичного включения обладает свойством симметричности и, следовательно, отражает взаимосвязь R -систем одного уровня.

Тогда на основе отношений полного и частично вхождения можно определить на множестве всех RS^i единственную R -систему с сетевой структурой в виде:

$$\overline{RS} = \cup RS^i = \{OS\} = \{ \langle m, r, \emptyset \rangle \mid m \subseteq U, r \subseteq R \}.$$

Отношение включения, отражая основные (наиболее устойчивые) взаимосвязи между R -системами, позволяет формировать OS , принадлежащие только одной RS^i , а отношение неполного включения дает возможность $OS^i \in RS^i$ приобретать новые свойства путем добавления новых первичных элементов, новых отношений между элементами и изменения законов композиции, что позволит соотнести OS^i с $OS^j \in RS^j$. Если же в RS^j будет формироваться набор OS^i со свойствами, выходящими за пределы RS^i , и в \overline{RS} не найдется RS^i , OS^i которой соответствовали бы OS^j , то в \overline{RS} должна

быть сформирована новая R -система, позволяющая объединить OS^i с новыми свойствами.

В соответствии с ОТС-У в OS необходимо выделять отношения, формирующие подсистемы эволюционных и неэволюционных преобразований, отношений противоречия и непротиворечия, полиморфизации и изоморфизации, симметрии и асимметрии, эволюции (изменения) и сохранения, позволяющие относить OS к различным R -системам. R -системы различных типов формируют в \overline{RS} самостоятельные подсистемы на основе отношения включения.

Таким образом, все допустимые изменения и эволюция OS всегда происходят в рамках некоторой R -системы, даже если начальная OS^i и конечная OS^j принадлежат разным R -системам, что определяется их замкнутостью относительно единственной системы \overline{RS} .

Основываясь на теоретических положениях и системном методе ОТС-У, разрабатываем модель ПСД.

Для этого сначала определим основные уровни организации СД: конструктивный элемент, узел, подструктура, структура. При этом будем учитывать наличие двух типов структур данных: абстрактных (логических), ориентированных на адекватное отображение реального мира, и конкретных (физических), обеспечивающих надежную и эффективную обработку данных на компьютере.

Конструктивные элементы являются минимальными компонентами СД и включают: элементы данных, параметры, признаки и указатели.

Конструктивные элементы логических СД представляют собой абстрактные элементы, для которых может быть задан абстрактный тип данных.

Абстрактный тип данных (АТД) задает множество значений данных D и операций над ними Ω и не учитывает представление данных и реализацию операций в компьютере [5, 6]. Формально абстрактный тип данных задается в виде:

$$t_g = \langle D_g, \Omega_g \rangle$$

где t_g – абстрактный тип данных с именем g ; D_g – множество значений типа g ; Ω_g – множество операций, выполняемых над значениями типа g .

В общем случае в АТД в качестве операндов операций могут выступать не только данные, определенные АТД, но и данные других типов: стандартных типов языка программирования или определенных в других АТД. Результат операции также может иметь тип, отличный от определенных в данной спецификации АТД.

На физическом уровне конструктивные элементы задаются с помощью конкретных (физических) элементов, для которых определен тип данных, реализуемый на компьютере (в языке программирования).

Будем называть **R -системой 1-го уровня** множество допустимых конструктивных элементов определенного типа, которые будем считать атомарными (неделимыми) на данном уровне:

$$OS^1 = \langle m^1, \emptyset, \emptyset \rangle = \langle m^1 \rangle,$$

$$RS^1 = \{OS^1\} = \{ \langle m^1 \rangle \mid m^1 \subseteq M^1 \} \equiv \{ \langle m^1 \rangle \mid m^1 \in M^1 \},$$

где $A^1 = g_1^1$ – основания выделения конструктивных элементов типа g_1^1 ; $m^1 \in M^1$; $R^1 = \emptyset$; $Z^1 = \emptyset$.

Примерами R -систем 1-го уровня являются множества элементов данных D , параметров L , признаков C и указателей U :

$$OS_1^1 = \langle d \rangle, RS_1^1 = \{ \langle d \rangle \mid d \in D \},$$

$$OS_2^1 = \langle l \rangle, RS_2^1 = \{ \langle l \rangle \mid l \in L \},$$

$$OS_3^1 = \langle c \rangle, RS_3^1 = \{ \langle c \rangle \mid c \in C \},$$

$$OS_4^1 = \langle u \rangle, RS_4^1 = \{ \langle u \rangle \mid u \in U \}.$$

На физическом уровне OS^1 и RS^1 могут быть представлены в виде:

$$OS_1^1 = \langle 5 \rangle, RS_1^1 = \{ \langle d \rangle \mid d \in D_{Integer} \} \text{ и } 5 = d_1,$$

где 5 – значение переменной, $D_{Integer}$ – множество значений целого типа данных $Integer$.

Сильно связанные элементы группируются в узлы (блоки).

Узлом называется сильно связанная система (композиция) конструктивных элементов, обычно строящаяся по единым законам и предоставляющая единый метод доступа к элементам.

Назовем R -системой 2-го уровня множество узлов, содержащее все возможные сочетания конструктивных элементов:

$$OS^2 = \langle m^2, r^2, z^2 \rangle,$$

$$RS^2 = \{ OS^2 \} = \{ \langle m^2, r^2, z^2 \rangle \mid m^2 \subseteq M^2, r^2 \subseteq R^2, z^2 \subseteq Z^2 \},$$

где $M^2 = \{ OS_1^1, OS_2^1, \dots, OS_n^1 \}$; n – число конструктивных элементов в узле; OS_i^1 – конструктивный элемент типа g_i^1 ; $A^2 = [g_1^1, g_2^1, \dots, g_n^1]$ – основания для выделения конструктивных элементов соответствующих типов; R^2 – отношение композиции (агрегации); Z^2 – ограничения, определяющие допустимое разнообразие объектов OS^2 в системе объектов RS^2 (разрешения и запреты на сочетание элементов в узле, ограничения на минимальное и максимальное количество элементов в узле и др.).

В таблице 1 приведены возможные комбинации конструктивных элементов для построения узлов. Приведенные композиции содержат только неповторяющиеся типы конструктивных элементов в одном узле. Поскольку типов элементов СД всего 4 (элементы данных, параметры, признаки и указатели), то число таких композиций равно:

$$\sum_{i=1}^4 C_4^i = 2^4 - 1 = 15.$$

Таблица 1

В системе объектов 2-го уровня можно выделить более мелкие подсистемы, например, системы объектов бинарных или тернарных узлов.

В общем случае узел СД может содержать произвольное количество конструктивных элементов.

В таблице 2 приведены возможные композиции узлов, содержащие различное количество элементов данных d и указателей u . Между элементами узла установлены связи, являющиеся фиксированными (позиционными) указателями.

Таблица 2

	0	1	...	n
0			...	
1			...	
2			...	
...
m			...	

Легко видеть, что число различных композиций равно $m \cdot n$, где m – максимальное число указателей, а n – максимальное число элементов данных в узле.

Вектор фазовых координат, характеризующий состав конструктивных элементов узла, зададим в виде (x_d, x_p, x_c, x_u) , где x_d – число элементов данных, x_p – число параметров, x_c – число признаков и x_u – число указателей в узле. Так, состав узла односвязного списка может быть задан с помощью вектора $(1, 0, 0, 1)$, а узла двусвязного списка – $(1, 0, 0, 2)$.

Узел бинарного дерева содержит элемент данных и два указателя на узлы-последователи и может содержать указатель на предшествующий узел. Состав этого узла определяется вектором $(1, 0, 0, 2)$ или $(1, 0, 0, 3)$. Аналогично в узле m -арного дерева (например, В-дерева) может храниться не более $m-1$ элементов данных и не более m указателей, поэтому состав узла определяется вектором $(m-1, 0, 0, m)$.

Узлы СД могут иметь равное количество конструктивных элементов, но при этом будут отличаться друг от друга их размещением (перестановками). Такие узлы называются изомерами. Например, изомерами являются: узлы $\langle d_1, d_2, u_1, u_2 \rangle$ и $\langle d_1, u_1, d_2, u_2 \rangle$, где d_1, d_2 – элементы данных, u_1, u_2 – указатели; узлы $\langle d, u \leftarrow, u \rightarrow \rangle$ и $\langle d, u \downarrow, u \uparrow \rangle$ – где d – элемент данных, $u \leftarrow, u \rightarrow$ – одноуровневые указатели на предшественника и последователя, $u \downarrow$ – иерархический указатель на потомка.

Различие между узлами можно обнаружить в форме представления элементов данных, например, в краткой или полной форме, в сжатом виде, в различных системах счисления, с различными единицами измерения и др.

В композиции узлов могут быть включены параметры, позволяющие строить СД из узлов одинаковой

структуры, но с различным количеством элементов, и признаки, позволяющие строить гетерогенные СД из узлов различной структуры.

Структура данных – это система, строящаяся по определенным законам из конструктивных элементов, узлов и подструктур.

Подструктура – это система, являющаяся структурно-функциональной подсистемой сложной структуры данных и строящаяся по определенным законам из конструктивных элементов, узлов и подструктур более низкого уровня.

Если всей СД соответствует граф, то подструктурам будут соответствовать собственные подграфы. Вследствие определенной обособленности подструктура может рассматриваться как элемент сложной СД.

Будем называть **R-системой 3-го уровня** множество структур:

$$OS^3 = \langle m^3, r^3, z^3 \rangle,$$

$$RS^3 = \{OS^3\} = \{ \langle m^3, r^3, z^3 \rangle \mid m^3 \subseteq M^3, r^3 \subseteq R^3, z^3 \subseteq Z^3 \},$$

где $M^3 = \{OS_1^1, OS_2^1, \dots, OS_n^1, OS_1^2, OS_2^2, \dots, OS_k^2, OS_1^3, OS_2^3, \dots, OS_q^3\}$; n, k, q – количество конструктивных элементов, узлов и подструктур в структуре; OS_j^1 – конструктивный элемент типа g_j^1 ; OS_j^2 – узел типа g_j^2 ; OS_j^3 – подструктура типа g_j^3 ; $A^3 = [g_1^1, g_2^1, \dots, g_n^1, g_1^2, g_2^2, \dots, g_k^2, g_1^3, g_2^3, \dots, g_q^3]$ – основания для выделения конструктивных элементов, узлов и подструктур соответствующих типов; R^3 – структурные отношения между элементами структуры; Z^3 – ограничения на структурные отношения.

Структурные отношения R^3 задают устойчивые (логические) взаимосвязи между элементами (компонентами) СД. В качестве примера таких взаимосвязей могут выступать отношения непосредственного следования (направления переходов от одних элементов к

другим), строго и нестрого порядка, иерархии, тождества, эквивалентности, функциональности и др. Так, доступ к элементам односвязного списка осуществляется последовательно, начиная с начала списка. Элементы двусвязного списка можно последовательно просматривать в прямом или обратном порядке.

Узлы деревьев связаны отношением иерархии от корня, являющегося входом в структуру, до конечных узлов – листьев, не имеющих потомков. Просмотр элементов возможен только от родительских узлов к узлам-потомкам.

В СД, реализованных с помощью связных списков, структурные отношения реализуются указателями. В массивах реализация отношений основана на преобразовании индекса (позиции) элемента в адрес физической памяти. Поэтому массивы предоставляют прямой доступ к элементам.

Законы композиции Z^3 – это ограничения на структурные отношения. Математически эти ограничения можно выразить с помощью предикатов. Например, условие сбалансированности АВЛ-дерева требует, чтобы для каждого узла высота его правого поддерева отличалась от высоты левого не более чем на единицу.

В бинарном дереве поиска значение ключа в любом узле должно быть больше либо равно значениям ключей во всех узлах левого поддерева этого узла и меньше либо равно значениям ключей во всех узлах правого поддерева этого узла:

$$x \leq k \leq y,$$

где k – значение ключа произвольного узла K ; x – значение ключа любого узла левого поддерева узла K ; y – значение ключа любого узла правого поддерева узла K .

В m -арном дереве поиска должно выполняться условие:

$$s_0 \leq k_0 \leq s_1 \leq \dots \leq k_{m-1} \leq s_m,$$

где k_0, \dots, k_{m-1} – значения ключей узла K , s_0, \dots, s_m – максимальные значения ключей соответствующих поддеревьев узла K .

В СД также могут накладываться ограничения на количество элементов, тип и число связей между ними, уникальность (наличие или отсутствие дубликатов) элементов и др.

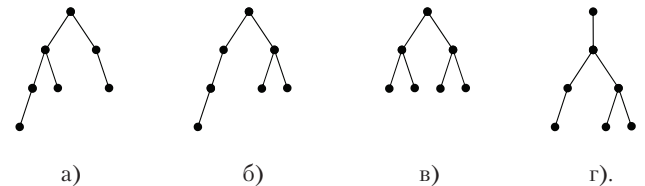
Операции, выполняемые над СД, должны сохранять структурные отношения в рамках законов композиции. Удовлетворение этих ограничений осуществляется с помощью специальных методов. Например, после вставки (удаления) нового элемента в сбалансированное дерево выполняется процедура балансировки, реализуемая посредством вращений.

Внутри системы объектов 3-го уровня могут выделяться подсистемы, формирующие отдельные виды, рода, классы, типы и др. Например, среди логических СД можно выделить следующие классы: множества (множества, мультимножества), последовательности (списки, списки с пропусками, циклические списки, очереди, стеки, деки), деревья (бинарные, тернарные, m -арные, произвольной арности), сети, графы.

СД также свойственно явление изомерии. Структуры-изомеры отличаются связностью элементов (топологией графа связей). Изомерами являются, например, различные классы бинарных деревьев, отличающиеся условием сбалансированности: АВЛ-деревья, красно-черные деревья, деревья Фибоначчи. Структуры-изомеры могут относиться к разным типам СД, например, двусвязные списки и бинарные деревья, узлы которых содержат элемент данных и два указателя.

Явление изомеризации СД объясняется тем, что количество конструктивных элементов и других инвариантов (количество смежных элементов, число компонентов связности и т.д.) не определяют граф связей СД с точностью до изоморфизма.

Ниже приведены графы, соответствующие различным вариантам построения бинарного дерева:



Графы не изоморфны друг другу, поскольку в общем случае не существует биекции, сохраняющей смежность вершин. Однако между вершинами графов можно установить взаимно однозначное соответствие. Поэтому каждый из графов может быть преобразован в другой с помощью некоторого набора операций перестановки. При этом топологическими свойствами (инвариантами) приведенных графов являются число вершин (7) и ребер (6), максимальная валентность вершин (2), число компонентов связности (1) и др.

Таким образом, представленные СД являются изомерами. Однако они будут обладать различными особенностями обработки данных. Например, известно, что высота сбалансированного бинарного дерева ограничена значением $2\log_2 n$, где n – количество элементов в дереве. Сбалансированные деревья ориентированы на равновероятный доступ к элементам: время доступа к данным и сложность их обработки пропорциональны $O(\log_2 n)$.

Несбалансированные деревья имеют большее число иерархических уровней, чем сбалансированные. Однако они могут быть организованы с учетом интенсивности (частоты) использования данных: наиболее часто используемые данные будут располагаться на верхних уровнях иерархии, а редко используемые – на нижних.

Основываясь на рассмотренных теоретических обоснованиях и организации СД сформулируем формальное определение поля структур данных.

Пространство всех потенциально возможных СД с заданными на нем бесконечными множествами конструктивных элементов, структурных отношений между конструктивными элементами и законов ком-

позиции, структурно совпадающее с \overline{RS} , будем называть **полем структур данных**.

СД, создаваемые и используемые в программах, являются функциональными системами [7], т.е. целостными объектами, способными самостоятельно функционировать в некоторой среде. Реально существующие СД в процессе своего существования претерпевают различные изменения, что позволяет сопоставлять их состояниям точки ПСД.

Основываясь на вышеизложенном, можно показать соответствие и полноту преобразований СД в ПСД преобразованиям объектов-систем, определенным в ОТС-У.

Теорема. Возможны лишь 4 основных и 11 производных способов преобразования СД.

В ОТС-У установлено, что существуют лишь 4 основных преобразования объекта-системы в рамках системы объектов одного и того же рода: тождественное, количественное, качественное, относительное и 11 производных преобразований, полученных комбинацией основных преобразований. Поэтому необходимо показать, что различные СД, построенные из элементов данных, параметров, признаков и указателей позволяют отразить все возможные модификации СД. Для этого сопоставим элементы СД с основными преобразованиями объекта-системы в ОТС-У.

Учитывая, что СД предназначена для организации и хранения данных, то тождественное преобразование ОТС-У будет соответствовать различным формам представления элементов данных, например, в краткой или полной форме, в сжатом виде, в различных системах счисления, с различными единицами измерения и др.

Использование в СД параметров соответствует количественным преобразованиям системы в ОТС-У, т.к. позволяет зафиксировать максимальный размер элемента, максимальное количество элементов в узлах и другие количественные ограничения на значения элементов данных и отношения между ними.

Наличие признаков в СД соответствует качественному преобразованию системы в ОТС-У, т.к. позволяет зафиксировать различные особенно-

сти отдельных элементов данных, узлов, подструктур и СД в целом, или информировать одни элементы о свойствах других элементов.

Использование указателей соответствует относительному преобразованию системы в ОТС-У, т.к. изменяет способ доступа между элементами СД, выделяя отдельные элементы или группы элементов с определенными свойствами.

Изменения СД с использованием различных комбинаций форм представления данных, параметров, признаков и указателей соответствуют производным преобразованиям системы в ОТС-У.

Теорема доказана.

Таким образом, ПСД потенциально позволяет создавать любые известные или мыслимые СД и допускает их различные преобразования согласно ОТС-У. Следовательно, ПСД представляет собой поле возможных путей эволюции реальных СД. Эволюция конкретной СД представляется непересекающейся траекторией в ПСД, в которой последовательность точек отражает последовательные изменения состояний СД в результате выполнения операций обработки данных.

СПИСОК ЛИТЕРАТУРЫ

1. Дрождин В.В., Володин А.М. Синергетический подход к организации структур данных // Программные продукты и системы. 2010. № 1. С. 29–34.
2. Словарь по кибернетике / Под редакцией В.С. Михалевича. Киев: Гл. ред. УСЭ им. М.П. Бажана, 1989. 751 с.
3. Система. Симметрия. Гармония / Под ред. В.С. Тютютина и Ю.А. Урманцева. М.: Мысль, 1988. 315 с.
4. Урманцев Ю.А. Эволюционика, или общая теория развития систем природы, общества и мышления. М.: Книжный дом «ЛИБРОКОМ», 2009. 240 с.
5. Седжвик Р. Фундаментальные алгоритмы на С++: анализ, структуры данных, сортировка, поиск. Киев: ДиаСофт, 2001. 688 с.
6. Нагао М., Катаяма Т., Уэмура С. Структуры и базы данных. М. Мир, 1986. 197 с.
7. Абдеев Р.Ф. Философия информационной цивилизации. М.: ВЛАДОС, 1994. 336 с.