4. Гермейер Ю.Б. Введение в теорию исследования операций. М, "Наука", 1971,

5. Балакирев В.С. и др. Об оценке эффективности АСУ научного эксперимента. "Приборы и системы управления", 1975, № 3.

6. Болтянский А.А., Васин Н.Н., Секисов Ю.Н., Скобелев О.П. Коммутационное преобразование напряжения низкого уровня. "Измерительная техника", 1974, № 4 с. 34-37.

УДК 681.3

В.В.Куликов

О СТРУКТУРЕ ДАННЫХ В СИСТЕМЕ АВТОМАТИЗИРОВАННОЙ ОБРАБОТКИ ИНФОРМАЦИИ

При разработке систем автоматизированной обработки данных на ЦВМ возникает задача описания и представления данных, образующих информационные потоки данных системы. Методы описания данных, используемые при разработке отдельных систем обработки [1,2], плохо применимы при описании данных, поступающих с измерительных систем, используемых при натурных и стендовых испытаниях сложных технических объектов. Одной из особенностей таких измерительных систем является частая смена комбинаций и числа измеряемых параметров и их представления. Внесение изменений в систему обработки данных при изменении состава измеряемых параметров может быть упрощено, если имеется возможность их нормального описания.

Пусть $X=\{x_1, x_2, \dots x_n\}$ множество элементарных данных, составляющих некоторую запись в ЦВМ, поступающую в систему обработки. На различных этапах обработки этой записи требуется выделить некоторые совокупности элементарных данных. Для определенности будем предполагать, что запись представляет входной сигнал $x(t)$, имеющий некоторый набор характеристик $x_i \in X_i /_{i=\overline{1,m}}$, где X_i - множество значений характеристики x_i при заданном времени t. Сигнал $x(t)$ можно представить в виде [3], $x=(x_1, x_2, \dots, x_n)$, где x_i - элементарные составляющие входного сигнала x;

$$(x_j=t, \ 1 \leq j \leq n).$$

Прямое произведение $\hat{X}=X_1 \times X_2 \times \dots \times X_m$ задает пространство входного сигнала. Множества $X_i /_{i=\overline{1,m}}$ - элементарные оси этого пространства. Входной сигнал x представляет собой точку $\hat{x}$ пространства $\hat{X}$ с координатами $x_1, x_2, \ldots, x_m$. Каждая характеристика x_i сигнала x является проекцией точки $\hat{x}$ на соответствующую ось X_i. При обработке сложной записи, представляющей сигнал x, удобно использовать обобщения понятия оси и проекции.

Пусть некоторые совокупности элементарных данных образуют составные данные. При этом в пространство элементарных данных необходимо добавить новую ось, соответствующую понятию составного данного. Обозначим X^*_{j1} пространство, образованное осями $X_1, X_2, \ldots, X_p$, то есть $X^*_{j1} = X_1 \times X_2 \times \ldots \times X_e$ или короче $X^*_{j1} = \Pi(X_i \in \{X_i\}_{i=\overline{1,e}})$.

Пространство X^*_{j1} можно рассматривать тоже как ось пространства $\hat{X}$. Проекция точки $\hat{x}$ на эту ось будут значения $x_1, x_2, \ldots x_e$.

Множества X_i, не являющиеся прямыми произведениями каких-либо множеств,- это элементарные оси пространства $\hat{X}$, в отличие от осей X_i^*, являющихся прямыми произведениями множеств $X_i \subset \hat{X}$. Таким образом, оси, не являющиеся "сомножителями" в выражении прямого произведения $\hat{X}$, можно рассматривать в качестве осей этого пространства, выделяя тем самым только необходимые значения сложного данного, участвующего в обработке.

В системе автоматизированной обработки данных, являющейся алгоритмической системой управления [4] частного типа, описание данных производится с помощью алгоритмических языков. Многие формальные языки, и, в частности, алгоритмические языки, используемые для целей программирования, описывают конструкции (операторы, данные, выражения и т.п.) цепочек символов системой составляющих [5]. Система составляющих является удобной моделью представления высказываний в формальных языках. Существенным для целей обработки данных является тот факт, что представление данных в некотором алфавите с использованием понятия системы составляющих, позволяет построить грамматику G, порождающую цепочки символов, представляющих эти данные. При некоторых ограничениях на свойства порождающей грамматики G, непосредственно по порождающей грамматике можно построить эффективный алгоритм анализа, проверяющей корректность данных [6].

Таким образом, понятие системы составляющих, примененное к описанию синтаксической структуры данных, позволяет формально описывать данные, указав соответствующую порождающую грамматику G , а также реализовать по ней синтаксический анализатор. Необходимость формального описания структуры данных ощущается как на этапе проектирования системы, так и при ее последующих модификациях и эксплуатации.

Определим понятие структуры данных, определив вначале некоторые связанные со структурами понятия.

Пусть на множестве элементарных данных $X=\{x_1, x_2, \ldots, x_n\}$, образующих некоторую запись, задано разбиение множества X на n классов эквивалентности $X_i \subseteq X \, / i=\overline{1,n}$ [7] . При этом будут выполнены соотношения

$$X=\bigcup_{r=1}^{n} X_r ; \quad X_r \cap X_k=\varnothing; \quad r \neq k; \quad r, k=\overline{1, n}$$

Для элементов, принадлежащих одному классу эквивалентности выполняются условия :

1) $\forall (x_i \in X_r)(x_i R x_i)$ - рефлексивность,

2) $\forall (x_i, x_j \in X_r)(x_i R x_j \Rightarrow x_j R x_i)$ - симметричность,

3) $\forall (x_i, x_j, x_g \in X_r)(x_i R x_j \wedge x_j R x_g \Rightarrow x_i R x_g)$ - транзитивность.

В задачах обработки данных классу эквивалентности соответствует понятие типа данного (переменной). Тип устанавливает некоторые характеристики данного, например :

тип определяет класс значений данного,

каждое значение может принадлежать только одному типу,

тип определяет набор допустимых операций,

тип данного позволяет решать вопрос о правильности структурных данных, а также вопрос о представлении данных и допустимых преобразованиях.

Структурные данные можно рассматривать как данные составного типа. Для n типов данных условие принадлежности элементов составному типу можно получить следующим образом. Отношение R , являющееся отношением эквивалентности на множестве,- это пара $\langle X, R \rangle$, $R \subseteq X \times X$. Прямая сумма отношений $\{\langle X_i, R_i \rangle\}_{i=\overline{1,n}}$ для классов эквивалентности $X_i \, / i=\overline{1,n}$ - это пара $\langle \bigcup_{i=1}^{n} X_i, \bigcup_{i=1}^{n} R_i \rangle$. Обозначим $R=\bigcup_{i=1}^{n} R_i$, $X=\bigcup_{i=1}^{n} X_i$. Для выполнения соотношения $x_i R x_j$ (т.е. $x_i, x_j \in X$) необходимо обеспечить выполнение условий :

I) $\{\forall(x_i, x_j \in \bigcup\limits_{r=1}^{n} x_r)(x_i R x_j)\}_{r=\overline{1,n}}$;

2) $\forall(r,g \in N)(X_r \cap X_g = 0)|_{r \neq g;\ r,g=\overline{1,n}}$.

Существуют и другие операции над эквивалентностями, дающие в результате эквивалентность.

На множестве $X=\{x_1, x_2, \ldots, x_n\}$ можно задать отношение строгого порядка $\angle$. Отношение $\angle$ обладает свойствами антирефлексивности и транзитивности :

I) $\forall(x_i, x_j \in X)(x_i \angle x_j \Rightarrow x_i \neq x_j)$

2) $\forall(x_i, x_j, x_k \in X)(x_i \angle x_j \wedge x_j \angle x_k \Rightarrow x_i \angle x_k)$

Из антирефлексивности отношения $\angle$ следует его антисимметричность

$$\forall(x_i, x_j \in X)(x_i \angle x_j \wedge x_j \angle x_i \Rightarrow x_i = x_j),\ i,j=\overline{1,K}$$

Множество X с заданным на нет отношением строго порядка $\angle$, т.е. пару $\langle X, \angle \rangle$ называют упорядоченным множеством и обозначают $\langle X, < \rangle$. Если на конечном множестве X задано отношение $<$, то на $X=\{x_1, x_2, \ldots, x_n\}$ можно выбрать такую нумерацию, что соотношение $x_i < x_j$ будет выполнено только в случае, когда $i=j$.

Отношением частичного порядка или нестрогим порядком является отношение $\angle^* = \angle \cup E$, где E – диагональное отношение множества X^2 , а $\angle$ – отношение строгого порядка. Для частичного порядка выполняются условия :

I) $\forall(x_i \in X)(x_i \angle^* x_i)$ – рефлексивность,

2) $\forall(x_j, x_i \in X)(x_i \angle^* x_j \Rightarrow \neg x_j \angle x_j)$ – антисимметричность,

3) $\forall(x_i, x_j, x_k \in X)(x_i \angle^* x_j \wedge x_j \angle^* x_k \Rightarrow x_i \angle^* x_k)$ – транзитивность.

Частичная упорядоченность обозначается $\leqslant$. Элементы $x_i, x_j \in X$ такие, что $x_i \leqslant x_j$ или $x_i \geqslant x_j$ называют сравнимыми элементами частично упорядоченного множества. Частично упорядоченное множество может содержать несравнимые элементы.

Линейно упорядоченное множество – это упорядоченное множество с попарно сравнимыми элементами $x_i \in X$.

Пусть X – частично упорядоченное множество, Y – некоторое подмножество множества $X, Y \subset X$. Элемент $x_i \in X$ такой что $x_i \geqslant x_j$ для любого $x_j \in Y$ называется верхней гранью подмножества Y. Наибольшим элементом упорядоченного множества X называется элемент $x_i \in X$, являющийся верхней гранью самого множества X. Аналогично определяется нижняя грань подмножества - $Y \in X$ и наименьший элемент множества X.

Элемент множества $x_{max} \in X$ называется максимальным в множестве X, если не существует $x_i \in X$, такое, что $x_{max} < x_i$. Аналогично определяется максимальный элемент $x_{min} \in X$. Множество X может не обладать ни наименьшим, ни наибольшим элементами и в то же время обладать одним или несколькими максимальными (минимальными) элементами, при этом максимальные (минимальные) элементы попарно несравнимы, а множество X частично упорядоченно.

Наименьший элемент множества верхних граней подмножества $Y \subset X$ называется точной верхней гранью множества и обозначается $Sup_x Y$. Точная нижняя грань определяется по аналогии с верхней гранью и обозначается $Inf_x Y$.

Элементы $x_i, x_j \in X$ при $x_i \leqslant x_j$ определяются в частично упорядоченном множестве X; подмножество $\{x_i, x_j\}$ такое, что $x \in \{x_i, x_j\}$ лишь в случае, когда $x_i \leqslant x \leqslant x_j$. Подмножество $\{x_i, x_j\} \in X$ называется интервалом частично упорядоченного множества X.

Интервал называется простым, если включает лишь свои концы x_i и x_j.

Частично упорядоченное множество X называется структурой (или решеткой), если для любой пары элементов $x_i, x_j \in X$ существует точная нижняя грань $Inf_x\{x_i, x_j\}$ и точная верхняя грань $Sup_x\{x_i, x_j\}$. Структура может быть задана в виде графа структуры.

С описанием синтаксической структуры цепочек символов связан класс порождающих грамматик [8]. Порождающая грамматика - это четверка $G = (V_T, V_N, S, P)$, где V_T - терминальный алфавит (словарь основных символов); V_N - нетерминальный алфавит (словарь вспомогательных символов), $V_N \cap V_T = \varnothing$; S - начальный символ (аксиома) грамматики G,

$S \in V_N$; $P = \{\xi_i \to \eta_i\}_{i=\overline{1,k}}$ - конечное множество правил подстановок, $\to \notin V = V_T \cup V_N$.

Алфавиты V_T и V_N предполагаются конечными. Левые и правые части правил подстановок $P_i : \xi_i \to \eta_i|_{i=\overline{1,k}}$ - это цепочки

символов алфавита $V = V_T \cup V_N$ - (цепочки над словарем V).
Правила $P_i \in P$ называют правилами вывода или схемой грамматики G .

Терминальные символы $x_i \in V_T / i=\overline{1,n}$ - элементарные символы определяемого языка $L(G)$.

Остальные символы, составляющие цепочки ξ_i, ζ_i , т.е. $A_j, S \in V_N / j=\overline{1,m}$ - метапеременные, используемые при выводе правильных цепочек.

Цепочка (строка) символов α прямо порождает цепочку β , если $\alpha = \gamma_1 \xi_i \gamma_2$ и $\beta = \gamma_1 \zeta_i \gamma_2$ и существует правило вывода $P_i: \xi_i \to \zeta_i \in P$, а γ_1, γ_2 - цепочки в алфавите $V_N \cup V_T$, возможно пустые. Кратко это обозначается $\alpha_G \to \beta$. Запись $\varphi_G \Rightarrow \psi$ означает, что существует последовательность цепочек $\omega_0, \omega_1, \ldots, \omega_n$ такая, что $\omega_0 = \varphi, \omega_n = \psi$ и $\omega_{i-1} \to \omega_i / i=\overline{0,n}$. Последовательность $\omega = (\omega_0, \omega_i, \ldots, \omega_n)$ называется выводом ω_n из $\omega_0 (\psi_{u_3} \varphi$, Каждому выводу ставится в соответствие последовательность правил вывода $P = (P_i, P_{i+1}, \ldots, P_{i+m}) / m=\overline{0,k}$, где $l = i+m$ - номер правила грамматики, использованного при выводе $\omega_{i-1} \to \omega_i$.

Между вхождениями некоторого символа в цепочку $\omega_{i-1} = \gamma_1 \xi_i \gamma_2$ и вхождениями этого же символа в цепочку $\omega_1 = \gamma_1 \zeta_i \gamma_2$ можно установить соответствие. Такие соответствующие друг другу вхождения символов в цепочку называются копиями символов в выводе. Выделенное вхождение символа x в цепочку ω_{i-1} , к которому на i -м шаге применялось правило вывода P_i , называется непосредственным предком всех вхождений символов, образующих подцепочку ζ в ω_i , а эти последующие вхождения - непосредственными потомками символа x . Понятия копий, непосредственных предков и непосредственных потомков обобщаются для случая несоседних цепочек вывода.

Каждому правилу $P_i: \xi_i \to \zeta_i$ можно сопоставить подграф (рис.I) в котором каждому символу x_i , входящему в цепочку $\xi_i = x_1, x_2, \ldots, x_n$ соответствует вершина, соединенная дугой с вершиной, помеченной номером правила P_i . Из этой вершины выходящие дуги, соединяют вершину P_i с вершинами, соответствующими символам, входящим в строку $\zeta_i = y_1, y_2, \ldots, y_m$

Указанное соответствие позволяет построить граф вывода цепочки ω . На графе вывода всем копиям символов соответствует одна вершина.

Таким образом, используя схему порождающей грамматики сос-

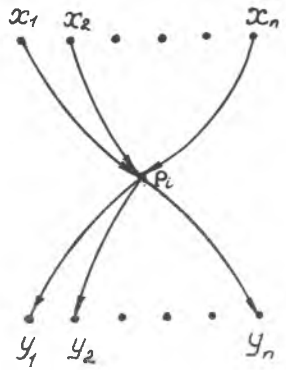

Рис. I.

тавляющих или HC - грамматики, можно построить граф вывода, являющийся графом структуры некоторой записи. По графу структуры можно найти все связи между элементарными данными, являющимися терминальными вершинами графа вывода. По этому графу можно выделить совокупности элементарных данных, образующих некоторое данное составного типа и т.д.

Каждое элементарное данное записи обладает только одним типом. Совокупность элементарных и составных данных будем называть составной переменной, в отличие от элементарной.

Каждая переменная x_i может иметь несколько значений $\overline{x_i}$, определяемых соответствующими проекциями на оси пространства входного сигнала $X(t)$. Область определения $\mu(x_i)$ переменной x_i - это множество допустимых значений $\overline{x}_i$ [9] . Число переменных (простых и составных), образующих переменную x_i , обозначим через $/x_i/ = n_i$.

С переменной x_i связаны ее тип и идентификатор. Тип переменной, простой или составной, используется для проверки границ значений и выбора семантической процедуры обработки. Идентификатор составной перменной связан с некоторым правилом p_i грамматики G , порождающей структуру. Для простой переменной идентификатор - это имя заключительной вершины структурного графа.

В настоящее время большое внимание уделяется развитию теории контекстно-свободных языков и применению алгебраических методов для изучения алгебраических свойств операций над КС - языками [I0] . Представляет интерес использование результатов этих

3-8254

работ для изучения свойств операций над структурными данных. Особенно актуально стоит эта задача в системах автоматизированной обработки данных измерительных систем, связанных с экспериментальными работами.

Дальнейшее развитие формального описания, повидимому, необходимо сочетать с поиском базовых структур, из которых можно строить на основе формальных алгебраических операций структуры данных требуемой сложности. В связи с разработкой банков данных такие работы интенсивно проводятся за рубежом [II, I2] . Представляет практический интерес изучение и реализация структур данных в вычислительной системе.

ЛИТЕРАТУРА

I. Барминский К.О., Губин А.И., Душкин Б.М. Операции над массивами и их программная реализация.-В сб. Труды НИИУМС , Пермь,I972, вып. УI.

2. Садовников В.И., Эпштейн А.Л. Потоки информации в системах управления. М., "Энергия", I974.

3. Бусленко Н.П., Калашников В.В., Коваленко И.Н. Лекции по теории сложных систем. М., "Советское радио", I973.

4. Тимофеев Б.Б. и др. Алгоритмизация в автоматизированных системах управления". Киев, "Техника", I972.

5. Гладкий А.В. Формальные грамматики и языки. М., "Наука", I973.

6. Шумей А.С., Зонис В.С. О синтаксическом анализе по однозначным грамматикам. "Программирование", I975, № 3.

7. Калужнин Л.А. Введение в общую алгебру. М., "Наука",I973.

8. Виленкин С.Я., Трахтенгерц Э.А. Математическое обеспечение управляющих вычислительных машин. М., "Энергия", I972.

9. Мельников И., Тамм Б. Об одном методе формализации структур данных. Изд. Академии Наук Эстонской ССР.Физика, Математика, I974, т. 23, № I.

I0. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра. Языки. Программирование. Киев, "Наукова думка", I974.

II. Алгебраическая теория автоматов, языков и полугрупп. Под. ред. М.А.Арбиба. Пер. с анг. М., "Статистика", I975.

I2. Информационные системы общего назначения. Аналитический обзор систем управления базами данных. Пер. с анг. М., "Статистика", I975.