

задачи, равного $i - 2M + 1$.

Л и т е р а т у р а

Г. Корбут А.А., Финкельштейн Ю.Ю.
Дискретное программирование. М., "Наука", 1969, 368 с.

В.В. Куликов, С.И. Трещев

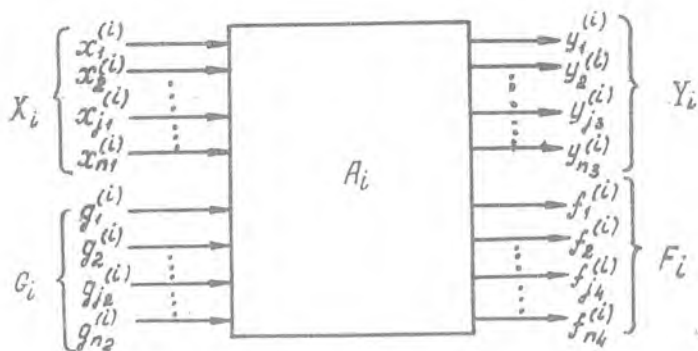
МОДЕЛИРОВАНИЕ ИНФОРМАЦИОННЫХ МАССИВОВ

При отладке и тестовой проверке системы программного обеспечения (СПО) измерительно-вычислительного комплекса (ИВК) возникает задача моделирования информационных массивов с заданной структурой элементов. В самом общем виде задача заключается в генерации цепочек (строк) символов при условии, что между совокупностями цепочек символов и между отдельными символами цепочек заданы отношения.

СПО ИВК имеет в своем составе ряд подсистем, обеспечивающих преобразование данных, поступающих с регистров устройств согласования ЦВМ с системой сбора данных, в форму, используемую при вторичной обработке.

Подсистему СПО можно рассматривать как алгоритмический преобразователь входной информации в выходную. Описание алгоритма преобразования задано алгоритмическим языком, на котором реализована совокупность программ подсистемы. Рис. 1 поясняет взаимодействие подсистемы A_i с информационными массивами.

Входной информационный массив X_i от подсистемы A_{i-1} ($i > 1$) или внешней среды (A_0) поступает в подсистему A_i . В процессе преобразования подсистема A_i использует информационный массив G_i , генерирует массив Y_i и обновляет часть данных массива F_i . Массивы G_i и F_i составлены из данных, не входящих непосредственно в очередной экземпляр массива X_i или Y_i . Например, массивы $G_i \cup F_i$ могут составлять константы, таблицы, номера алгоритмов обработки, списки обработанных параметров, логические шкалы и т.д., а массив $X_i \cup Y_i$ отдельные значения обрабатываемой реализации и другие необходимые для подсистемы A_i атрибуты параметра



Р и с. I.

(время, тип и т.п.).

Структура записей информационного массива X_i существенно зависит от функции подсистемы A_i . Например, информация на регистрах устройства согласования ЦВМ с измерительной системой закодирована в форме, принятой при разработке измерительной системы и устройств согласования. Одно слово такого регистра обычно содержит информацию о типе λ и значениях u параметра, времени t получения отсчета и координатах n отсчета в измерительной системе. Число значений, принимаемой каждой переменной λ , u , t и n зависит от мощности множеств Λ , U , T и N , на которых определены эти переменные. Порядок и число переменных в одном слове регистра может зависеть от типа слова λ и (или) типа параметра.

Такой информационный массив может быть преобразован в стандартный (для используемой ЦВМ) n -мерный массив с размерностью, равной максимальному числу переменных, упакованных в одно слово регистра. Подсистема формирования записей для выдачи информационного массива на печать генерирует символьные строки выходного документа в алфавите АЦПУ. Выходной документ чаще всего имеет вид таблицы или списка.

Разнообразие структур записей на входах различных подсистем СПО выдвигает также задачу удобного описания структур, данных с учетом формы их представления в памяти ЦВМ. Эту задачу можно сформулировать как двухэтапную. На первом этапе с помощью подязыка можно описать форму элементов информационных массивов и задать структуру записей. На втором этапе для всех

элементов можно задать форму их представления с учетом ограничений на параметры элемента. Например, можно задать структуру числа с фиксированной точкой с помощью формул Бэкуса (первый этап), а затем перевести это число в требуемую систему счисления, предварительно проверив допустимый диапазон числа и округлив, если это необходимо. После операции выравнивания числа на заданную границу внутри машинного слова, этот элемент можно поместить в соответствующую таблицу (второй этап).

При порождении информационного массива можно поступить аналогичным образом. Термин "моделирование" используется в данной работе, чтобы различать порождение цепочки символов, как предложения некоторого языка $L(G)$, от генерации последовательности символов с учетом семантических свойств порождаемого объекта. Таким образом, моделируемый информационный массив, независимо от формы представления его элементов в памяти ЦВМ вначале представляет собой цепочку символов алфавита, используемого при подготовке и отображении данных в ЦВМ. Преобразование элементов и совокупностей элементов в необходимую форму может быть выполнено с помощью семантических процедур, включаемых в описание структуры массива.

Сформулируем требования к алгоритму моделирования информационных массивов: этот алгоритм должен обладать способностью порождения (генерации) последовательностей элементов и информации о связях между элементами, если они заданы явно. Параметрами этого алгоритма является информация о структуре записей, ограничения на число символов элемента, требования к форме представления элементов в памяти ЦВМ.

Схему моделируемого информационного массива зададим продукциями порождающей KS -грамматики, включив в продукцию правила вывода с левой рекурсией для получения цепочек символов произвольной длины. Для генерации строк по порождающей грамматике используют два стека $INSTC$ и $OUTSTC$, два регистра RN и RS - для номера (индекса), применяемого правила и символа из стека, соответственно, и две одномерные таблицы VN и VT . В таблицах VN и VT содержатся символы множеств V_N и V_T порождающей грамматики $G = (V_T, V_N, S, P)$. Стек $INSTC$ выполняет функцию вспомогательной памяти. Стек $OUTSTC$ служит для накопления терминальной цепочки. Перед началом генерации строки в стек $INSTC$ заносится

символ \perp . Опишем схему алгоритма:

1. $RI = I$. Занести в $INSTC$ правую часть правила - аксиомы. Перейти к п.3;

2. Поместить в $INSTC$ правую часть правила с номером, хранящемся в RI ;

3. Занести верхний символ из $INSTC$ на регистр RS ;

4. Сравнить символ регистра RS с $S_i \in VN$. Если $S \in VN$, то занести номер правила для этого символа на RI и перейти к п.2;

5. Сравнить символ регистра RS с $S_i \in VT$. Если $S \in VT$, то занести этот символ S в $OUTSTC$ и перейти к п.3;

6. Сравнить символ регистра RS с символом " \perp ". Если $S \equiv \perp$ то закрыть $OUTSTC$.

7. Ошибка.

(При невыполнении условия какого-либо пункта $1 \leq i \leq \delta$ переходим к пункту $i+1$).

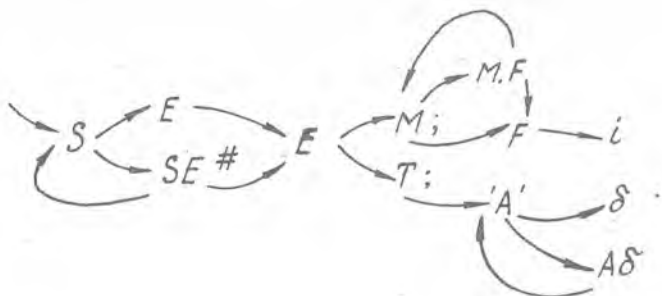
Поясним на примере процесс генерации строк по приведенному алгоритму. Пусть требуется моделировать набор записей, состоящих из последовательности текстов T и одномерных числовых массивов M . Зададим порождающую грамматику: $G_T = (V_T, V_N, S, P)$:

$V_T = \{;, ', \cdot, \sigma, \#, i\}$; $V_N = \{S, E, M, T, A, F\}$; $\notin V_N \cup V_T$;

$P = \{$	1. $S \rightarrow SE\#$	6. $M \rightarrow M.F$
	2. $S \rightarrow E$	7. $M \rightarrow F$
	3. $E \rightarrow M \cdot$	8. $A \rightarrow A\sigma$
	4. $E \rightarrow T$	9. $A \rightarrow \sigma$
	5. $T \rightarrow 'A'$	10. $F \rightarrow i\}$.

Граф структуры приведен на рис. 2. Состояния стеков и регистров при генерации строки приведены на рис. 3.

Представление данных в виде рабочих структур является второй частью задачи моделирования информационных массивов. Для этой цели можно применить анализатор [Л2], дополнив макрокоманды m_i $PM S, n_i$ ссылками на семантические процедуры обработки редуцируемых строк символов. Анализатор для грамматики G_T имеет вид:



Р и с. 2.

1. СР S , 9	14. БП $0ш$	27. СР $\#$, 37
2. СР E , 15	15. СР $\#$, 29	28. БП $0ш$
3. СР M , 17	16. БП $0ш$	29. ПМ S , 2
4. СР T , 20	17. СР $.$, 30	30. СР F , 38
5. СР $'$, 22	18. СР $;$, 32	31. БП $0ш$
6. СР F , 25	19. БП r $0ш$	32. ПМ E , 2
7. СР i , 26	20. СР $;$, 33	33. ПМ E , 2
8. БП $0ш$	21. БП $0ш$	34. СР $'$, 39
9. СР $E \neq$ 27	22. СР A , 34	35. БП $0ш$
10. СР M , 17	23. СР δ^0 , 36	36. ПМ A , 1
11. СР T , 20	24. БП $0ш$	37. ПМ S , 3
12. СР F , 25	25. ПМ M , 1	38. ПМ M , 3
13. СР i , 26	26. ПМ F , 1	39. ПМ T , 3

Дополнив макрокоманду 25-й процедурой накопления значений элементов массива M , одномерный массив $M = \{f_i\}_{i=1, n}$ - можно преобразовать в n -мерный массив, используя для этой цели n -стекную память.

Таким образом, используя схему порождающей грамматики и построенный на ее основе анализатор, можно задать структуру информационного массива и получить требуемое внутреннее представление элементов записей.

Для проверки возможности моделирования информационных массивов были применены языки ФМ-20, автокод и макрогенератор системы ОСПО ЭВМ М-222. Моделируемые объекты-массивы и простые переменные резервировались в языке Фортран. Для связи пунктов автокода и Фортрана был реализован оператор . типа

<i>N</i>	1 3 4 2 3 4 2 3 4 2 3 4 2 3 4 5 3 4 2 3 4 5	3 4 5
<i>I</i>	S E S E E # T ; , A , δ ...	E
<i>N</i>	E # E # # E : # A , : /	#
<i>S</i>	# L # E E # # / ; # ;	E
<i>T</i>	L E # # E E , # #	#
<i>C</i>	# L E # # #	L
	L # L E	
	L #	
	L	
<i>RS</i>	S S E T / A δ	
<i>RI</i>	1 1 2 4 5 9	
<i>O</i>	α	δ #
<i>U</i>		α / ;
<i>T</i>		α /
<i>S</i>		δ
<i>T</i>		;
<i>C</i>		δ
		/
		α
<i>RES</i>	α' δ δ ... δ' ; # o f, f, ..., f ; #	

Р и с. 3.

NAMELIST языка Фортран - IV ДЭС ЕС. Дополнительно был выяснена возможность использования арифметических и условных операторов, а также процедур языка ФМ-20.

Л и т е р а т у р а

1. Б е р а т и с с А.Т. Структура данных. Перевод с англ., М., "Статистика", 1974.
2. Ш у м е й А.С., З о н и с В.С. О синтаксическом анализе по однозначным грамматикам. М., "Программирование", 1975, №3.
3. Г р и с Д. Конструирование компиляторов для цифровых вычислительных машин. Перевод с англ., М., "Мир", 1975.