

## СЛИЯНИЕ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ЛОГИЧЕСКИХ ОПРЕДЕЛИТЕЛЕЙ

© В.И. Левин, А.В. Худяков

Levin V.I., Khudyakov A.V. Data confluence with the employment of logical determinants. The article looks at a method of presenting some simple operations on data in the information system. For this purpose, it is proposed to apply the apparatus of continuous logic and ordinal logical determinant in particular. Examples are supplied.

### ВВЕДЕНИЕ

Функционирование информационных систем предполагает выполнение множества операций над данными, содержащимися в них. Эти операции представляют собой некоторые типичные действия: включение новых данных в уже имеющуюся совокупность, исключение (удаление) данных из этой совокупности, поиск данных по его положению или значению, упорядочение их в каком-либо смысле (например, сортировка), объединение и разъединение (различные соединения, слияния, разбиения) и другие [1].

Существует множество методов и алгоритмов выполнения этих операций, каждый из которых применим для определенных структур данных. В данной статье рассматривается решение задачи представления операции слияния одномерных массивов с помощью аппарата непрерывной логики и логических определителей.

### 1. ПОСТАНОВКА ЗАДАЧИ

Задача слияния ставится следующим образом. Пусть имеется некоторое число  $n$  одномерных массивов различной длины  $m_i$ , вида

$$A_i = \{a_{i1}, a_{i2}, \dots, a_{im_i}\}, \quad i = \overline{1, n}, \quad (1)$$

где  $m_i$  – размерность массива  $A_i$ , т. е. количество его элементов. Элементы  $a_{ij}$  являются однотипными. Для каждого такого массива  $A_i$  определен порядок на множестве его элементов, известны операции поиска и упорядочивания элементов. Требуется образовать из всех элементов, принадлежащих всем массивам  $A_i$ ,  $i = \overline{1, n}$ , в совокупности единую структуру, в которой все элементы имели бы такой же порядок, что и в исходных массивах. Эта структура может быть названа результатом слияния массивов  $A_i$ .

### 2. КОНКРЕТИЗАЦИЯ УСЛОВИЙ

Пусть имеется совокупность массивов  $A_i$ , описанная в п. 1. Если элементы в  $A_i$  неупорядочены, то слияние таких массивов тривиально, оно производится про-

стым соединением или конкатенацией. Этот случай не представляет интереса.

Пусть теперь, как следует из условия п. 1, над элементами массивов задано отношение порядка. Установим одинаковое для всех  $A_i$  упорядочение элементов, а именно, по возрастанию. То условие, что упорядочение должно быть одинаковым, необходимо для обеспечения слияния. Операция слияния по своему характеру такова, что ее результат имеет те же свойства и структуру, что и аргументы. Это, в свою очередь, требует однородности массивов в указанном выше смысле. Конкретно, пусть элементы  $a_{ij}$  в каждом из  $A_i$  упорядочены по возрастанию по индексу  $j$ .

### 3. ИДЕЯ РЕШЕНИЯ

Примем условия упорядоченности и однородности, указанные в п. 2 для массивов  $A_i = \{a_{i1}, a_{i2}, \dots, a_{im_i}\}$ ,  $i = \overline{1, n}$ . После этого объединим их в матрицу  $A_n$  с  $n$  строками различной длины

$$A_n = \begin{pmatrix} a_{11} & \dots & a_{1m_1} \\ a_{21} & \dots & a_{2m_2} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nm_n} \end{pmatrix} \quad (2)$$

Образованная структура (2) может рассматриваться как одномерный массив, состоящий из  $m_1 + m_2 + \dots + m_n$  элементов.

При этом, т. к. все элементы однотипны и между ними априори существует отношение порядка, к такому массиву можно предъявить условие упорядочиваемости. Обычно это условие удовлетворяется существованием процедуры сортировки или поиска. В нашем случае упорядочиваемость гарантируется операцией поиска элемента с заданным номером  $r$ . Для реализации операции такого поиска применим аппарат непрерывной логики, а именно, понятие порядкового логического определителя [2]. В частности, значение логического определителя  $A_n^r$  ранга  $r$

$$A_n^r = \begin{vmatrix} a_{11} & \dots & a_{1m_1} \\ a_{21} & \dots & a_{2m_2} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nm_n} \end{vmatrix}^r \quad (3)$$

равно  $r$ -му по порядку возрастания элементу матрицы  $A_n$ . Вычисление логического определителя  $A_n^r$  проводится по формуле [2].

$$A_n^r = \bigvee_{\sum_{s=1}^n i_s = r+n-1} a_{1i_1} \wedge \dots \wedge a_{ni_n}, \quad (4)$$

где  $\vee$  – операция «max»,  $\wedge$  – операция «min» непрерывной логики.

Таким образом, в результате вышеописанных действий над матрицей  $A_n$  с различным числом элементов в строках, получается слияние векторов-строк данных  $A_i$  [3].

Вычисления  $r$ -го по порядку элемента можно проводить не нарушая структуры матрицы  $A_n$ , не требуя никаких перемещений данных, не привлекая для этого дополнительной памяти. Таким образом, имеем метод логического слияния строк без их физического изменения, т. е. с сохранением исходной последовательности элементов внутри них.

#### 4. ОЦЕНКА СЛОЖНОСТИ ВЫЧИСЛЕНИЙ

Наиболее трудоемкой частью производимых действий при вычислениях в рамках описанного в п. 3 подхода, является вычисление логического определителя  $A_n^r$ .

Оценка сложности вычислений логических определителей проведена в [4]. Полученные результаты показывают, что сложность зависит от количества элемен-

тов в строках определителя, количества строк и способа вычисления логических определителей. При наиболее оптимальном из них – блочной дихотомии, сложность растет линейно при увеличении ранга  $r$  и числа элементов  $n$  в определителе.

#### ЗАКЛЮЧЕНИЕ

Рассмотрена возможность реализации операции слияния структур данных в информационных системах, основанная на применении аппарата порядковой логики, в частности, логических определителей. Приведены данные о сложности необходимых вычислений.

Преимуществом такого способа реализации операции слияния является то, что элементы данных не перемещаются в памяти, т. е. слияние происходит только на логическом уровне. Особенно это выгодно в случае, когда данные имеют полилинейную структуру. Существенный недостаток подхода состоит в том, что для вычисления логического определителя требуется достаточно большое число арифметико-логических операций. Пути преодоления данного недостатка видятся в синтезе наиболее эффективных алгоритмов вычисления и поиска специальных случаев применения подхода, например, при большом числе определителей достаточно низкого ранга.

#### ЛИТЕРАТУРА

1. Конструирование систем программирования обработки данных. М.: Статистика, 1979. 270 с.
2. Левин В.И. Структурно-логические методы исследования сложных систем с применением ЭВМ. М.: Наука, 1987. 304 с.
3. Худяков А.В., Левин В.И. Применение порядковой логики для представления данных и операций над ними // Вестник УлГТУ. Ульяновск, 2000. (В печати).
4. Левин В.И. Бесконечнозначная логика в задачах кибернетики. М.: Радио и связь, 1982. 176 с.

Поступила в редакцию 21 декабря 2000 г.