Битовые шкалы

Пусть задан конечный универсум $\boldsymbol{U}$, и число элементов в нем не превосходит разрядности компьютера, $|U| < n$.

Элементы универсума пронумерованы:

$$U = \{u_1, u_2, \ldots, u_n\}.$$

Подмножество $\boldsymbol{A}$ универсума $\boldsymbol{U}$ представляется кодом (машинным словом или битовой шкалой) $\boldsymbol{C}$, в котором

$$C[i] = \begin{cases} 1, & \text{если } u_i \in A, \\ 0, & \text{если } u_i \notin A, \end{cases}$$

где $\boldsymbol{C}[\boldsymbol{i}]$ – $\boldsymbol{i}$-й разряд кода $\boldsymbol{C}$.

Термин «определение представления объекта» применительно к программированию означает следующее:

описание в терминах системы программирования

- структуры данных, используемой для хранения информации о представляемом объекте;
- алгоритмов над выбранными структурами данных, которые реализуют присущие данному объекту операции.

Как правило:

один и тот же объект может быть представлен многими разными способами, причём нельзя указать способ, который является наилучшим для всех возможных случаев.

Отличительная особенность хорошего программиста – знание множества способов представлений объектов и умение выбирать способ, наиболее походящий для решения данной задачи.

Битовые шкалы

Реализация операций над множествами в битовых шкалах.

- Код пересечения множеств $\boldsymbol{A}$ и $\boldsymbol{B}$ есть поразрядное логическое произведение кода множества $\boldsymbol{A}$ и кода множества $\boldsymbol{B}$.
- Код объединения множеств $\boldsymbol{A}$ и $\boldsymbol{B}$ есть поразрядная логическая сумма кода множества $\boldsymbol{A}$ и кода множества $\boldsymbol{B}$.
- Код дополнения множества $\boldsymbol{A}$ есть инверсия кода множества $\boldsymbol{A}$.

В современных компьютерах имеются соответствующие машинные команды → эти операции выполняются весьма эффективно

Битовые шкалы

<u>Пример</u>.

Пусть $U = \{1, 2, 3, 4, 5, 6, 7\}$,

$A = \{2, 3, 4, 6\}, \quad B = \{1, 3, 5\}.$

В этом случае

$\boldsymbol{C_A} = (0, 1, 0, 1, 1, 1, 0)$, $\boldsymbol{C_B} = (0, 0, 1, 0, 1, 0, 1)$.

Тогда

$$C_{A \cup B} = (0, 1, 1, 1, 1, 1, 1),$$
$$C_{A \cap B} = (0, 0, 0, 0, 1, 0, 0),$$
$$C_{\bar{A}} = (1, 0, 1, 0, 0, 0, 1).$$

<u>Замечание</u>.

Если мощность универсума превосходит размер машинного слова (но не очень велика), то для представления множеств используются массивы битовых шкал.

В этом случае операции над множествами реализуются с помощью циклов по элементам массива.

Алгоритм построения бинарного кода Грея

Во многих переборных алгоритмах требуется последовательно рассмотреть все подмножества заданного множества и найти среди них то, которое удовлетворяет заданному условию.

Работа алгоритма может быть значительно ускорена, если организовать его так, чтобы на каждом следующем шаге можно было использовать результаты, полученные на предыдущем шаге.

Алгоритм построения бинарного кода Грея

Описание алгоритма.

Вход: $n \ge 0$ – мощность заданного множества;

выход: последовательность кодов всех подмножеств заданного множества.

B – текущий элемент последовательности (битовый код длины n); последовательность содержит 2^n кодов.

$B[i]$ – i-й разряд текущего кода.

1. Инициализация:

для $i = 1, 2, \dots, n$

$B[i] = 0$

Формирование кода пустого множества

2. Вывод B.

3. Для $k = 1, 2, \dots, 2^n - 1$

3.1 $p = Q(k)$

Определение элемента, подлежащего добавлению либо удалению (вызов функции Q)

3.2 $B[p] = 1 - B[p]$

Добавление или удаление элемента (путем изменения соответствующего разряда битового кода)

3.3 Вывод B.

Описание функции Q.

Вход: k – номер подмножества;

выход: номер изменяемого разряда битового кода (количество нулей на конце двоичной записи числа k, увеличенное на 1).

1. $q = 1$; $j = k$

2. Пока j четно, выполнять

$j = j/2$; $q = q+1$

3. Возврат q

<u>Пример</u>.

Протокол работы алгоритма для $n = 3$.

1. $B = (0, 0, 0)$.

2. Вывод B.

3. $k = 1$

$p = Q(1) = 1$;

$B[1] = 1 - B[1] = 1 - 0 = 1$;

вывод $B = (0, 0, 1)$;

$k = 2$

$p = Q(2) = 2$;

$B[2] = 1 - B[2] = 1 - 0 = 1$;

вывод $B = (0, 1, 1)$;

…………

Результаты:

k	p	B
		(0, 0, 0)
1	1	(0, 0, 1)
2	2	(0, 1, 1)
3	1	(0, 1, 0)
4	3	(1, 1, 0)
5	1	(1, 1, 1)
6	2	(1, 0, 1)
7	1	(1, 0, 0)

Другие представления множеств

Если универсум очень велик или бесконечен, а рассматриваемые подмножества универсума не очень велики, то представление с помощью битовых шкал не является эффективным с точки зрения экономии памяти.

В этом случае множества представляются с помощью других структур:

- массивов,
- упорядоченных списков,
- коллекций (C#),
- ***set*** (Pascal) …

Прямое (декартово) произведение двух множеств

Пусть ***A*** и ***B*** – два множества.

Прямым (***декартовым***) ***произведением*** множеств ***A*** и ***B*** называется множество всех <u>упорядоченных</u> пар (***a***, ***b***), таких, что

$$a \in A, \quad b \in B.$$

Обозначение: ***A* × *B***.

Формально:

$$A \times B \overset{\text{def}}{=} \{(a, b) \mid a \in A \ \& \ b \in B\}.$$

Прямое (декартово) произведение двух множеств

<u>Пример</u>.

Пусть $A = \{a, b, c\}, \quad B = \{0, 1\}.$

Тогда

$$A \times B = \{(a,0), (a,1), (b,0), (b,1), (c,0), (c,1)\},$$
$$B \times A = \{(0,a), (0,b), (1,a), (1,b), (0,c), (1,c)\}.$$

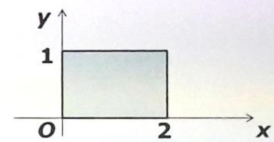
Как видно из примера, декартово произведение некоммутативно:

$$A \times B \neq B \times A.$$

Точка на плоскости может быть задана упорядоченной парой координат (т. е. двумя точками на координатных осях).

Таким образом, $R^2 = R \times R.$

В частности, если $X = [0, 2], \quad Y = [0, 1],$ то $X \times Y$ – множество точек прямоугольника.



Метод координат ввел в употребление Рене Декарт (1596 – 1650). Отсюда название «декартово произведение» (хотя теория множеств появилась более чем 200 лет спустя).

Мощность декартова произведения конечных множеств

Теорема.

Для конечных множеств ***A*** и ***B***

$$|A \times B| = |A| \cdot |B|.$$

В примере выше $A = \{a, b, c\}, \quad B = \{0, 1\},$

$$|A| = 3, \quad |B| = 2, \quad |A \times B| = 6.$$

Обобщение на большее число сомножителей

Прямым (***декартовым***) ***произведением*** множеств $\boldsymbol{A_1}, \boldsymbol{A_2}, \ldots \boldsymbol{A_n}$ называется множество всех упорядоченных наборов (*кортежей*):

$$A_1 \times A_2 \times \ldots \times A_n \overset{\text{def}}{=}$$
$$= \{(a_1, a_2, \ldots, a_n) \mid a_1 \in A_1 \ \& \ a_2 \in A_2 \ \& \ldots \& \ a_n \in A_n\}.$$

Степень множества

В определении декартова произведения сомножители не обязательно должны быть различными.

Степенью множества A называется его декартово произведение само на себя:

$$A^n \stackrel{\text{def}}{=} \underbrace{A \times A \times \ldots \times A}_{n\ \text{раз}}.$$

Для конечного множества ***A***

$$\left| A^n \right| = \left| A \right|^n.$$

Степень множества

Пример.

Пусть ***A*** – конечный алфавит (конечное множество, элементами которого являются символы).

Элементами множества A^n являются слова длины ***n*** в алфавите ***A***.

Под словом понимается любая последовательность символов данного алфавита.

Бинарные отношения

Пусть ***A*** и ***B*** – два непустые множества.

Бинарным отношением R между множествами A и ***B*** называется всякое подмножество декартова произведения ***A×B***:

$$R \subset A \times B.$$

Тот факт, что $(a, b) \in R$, т. е. между элементами $a \in A$ и $b \in B$ существует отношение ***R***, обозначается так: ***aRb***.