Х. РАМИЛЬ АЛЬВАРЕС

ПРОСТЫЕ АЛГОРИТМЫ ПЕРЕВОДОВ $p \to p-1$ и $p \to p+1$

Переводы чисел из p-ичной системы счисления в (p+1)-ичную (перевод $p \to p+1$) и в (p−1)-ичную (перевод $p \to p-1$) системы представляют интерес в связи с использованием троичной системы в ЦВМ [1] и при передаче информации в числовой форме, для которой использование троичного кода более экономно, чем двоичного [2]. Возникающие при этом переводы либо уже являются переводами рассматриваемого вида (например, переводы 2→3 и 3→2), либо могут быть сведены к этим переводам (например, переводы 10→3 и 3→10 могут быть сведены к переводам 10→9 и 9→10).

Будем рассматривать позиционные системы счисления с базисными числами от 0 до $p-1$, где p − основание системы. Представление числа N в p-ичной системе есть

$$(a_n \ldots a_0 . d_1 \ldots d_m) p,$$

где $0 \leq a_k \leq p-1$ и $0 \leq d_i \leq p-1$. При этом $(a_n \ldots a_0)p$ есть целая часть числа, а $(.d1 \ldots d_m)p$ − дробная часть.

Для целых чисел A и m ($m>1$) введем
$\{A\}m$ — наименьший неотрицательный вычет по основанию m;
$[A]m$ — определяемое из следующего соотношения:

$$A=[A]m_m+\{A\}m.$$

Правила перевода целых чисел основываются либо на вычислении в старой системе счисления частного и остатка от деления целого числа на новое основание, либо на вычислении в новой системе произведения целого числа на старое основание. Перевод дробных чисел базируется на вычислении в старой системе целой и дробной частей произведения дроби на новое основание. Поэтому рассмотрим алгоритмы деления и умножения целых чисел на $p-1$ и на $p+1$ в p-ичной системе.

1. Алгоритмы деления на $p-1$ и на $p+1$

Рассмотрим правила получения частного и остатка от деления целого числа N, имеющего представление $(a_n \ldots a_0)p$ на $p-1$ и на $p+1$.

1.1. Алгоритм деления на $p-1$.

Покажем, что если $a_n < p-1$, то

$$[(a_n \ldots a_0)p]_{p-1} = (bn\text{-}1 \ldots b_0)_p$$

$$\{(a_n \ldots a_0)_p\}_{p-1} = c_0$$

где $b_i (i=n-1 \ldots 0)$ и c_0 определяются по следующим рекуррентным формулам:

$$b_i = c_{i+1} + [a_i + c_i + 1]_{p-1}, \quad c_i = \{a_i + c_{i+1}\}_{p-1} \tag{1}$$

$$c_n = a_n$$

Для c_i, определяемых по формулам (1), справедливо неравенство $c_i < p-1$; отсюда $[a_i+c_i+1]_{p-1} \leq 1$ и $b_i < p$. Следовательно, $(b_{n-1}...b_0)$ есть представление некоторого числа в системе счисления с основанием p.

Справедливость формул (1) покажем методом индукции по n. Для n=1 имеем

$$a_1p + a_0 = a_1(p-1) + (a_0+a_1) =$$

$$= a_1(p-1) + (p-1)[a_0+a_1]p_{-1} + \{a_0+a_1\}_{p-1} =$$

$$= (p-1)(a_1 + [a_0+a_1]_{p-1}) + \{a_0+a_1\}_{p-1}$$

Отсюда $[(a_1a_0)]_{p-1} = a_1 + [a_0+a_1]_{p-1}$ и $\{(a_1a_0)_p\}_{p-1} = \{a_0+a_1\}_{p-1}$ т.е. для n=1 формулы (1) верны. Пусть формулы (1) верны для $n<k$, покажем, что они верны и для $n=k$:

$$\sum_{i=k}^{0} a_ip^i = a_kp^k + a_{k-1}p^{k-1} + \sum_{i=k-2}^{0} a_ip^i = p^{k-1}(a_kp + a_{k-p}) +$$

$$+ \sum_{i=k-2}^{0} a_ip^i = (p-1)(a_k+[a_{k-1}+a_k]_{p-1})p^{k-1} + \{a_{k-1}+a_k\}_{p-1}p^{k-1}+ \sum_{i=k-2}^{0} a_ip^i.$$

Так как формулы (1) верны для $n<k$, то для

$$\{a_{k-1}+a_k\}_{p-1}p^{k-1} + \sum_{i=k-2}^{0} a_ip^i \quad \text{имеем}$$

$$\{a_{k-1}+a_k\}_{p-1}p^{k-1} + \sum_{i=k-2}^{0} a_ip^i = (\mathrm{p}-1) \sum_{i=k-2}^{0} b_ip^i+c_0,$$

где $b_i(i=k-2...0)$ и c_0 определяются по формулам (1) и $c_{k-1} = \{a_k-1+a_k\}_{\mathrm{p}-1}$.
Таким образом, имеем

$$\sum_{i=k}^{0} a_ip^i =(p-1)(a_k+[a_{k-1}+a_k]_{p-1})p^{k-1} +$$

$$+ (p-1) \sum_{i=k-2}^{0} b_ip^i +c_0= (p-1) \left((a_k+[a_{k-1}+a_k]_{p-1})p^{k-1} + \sum_{i=k-2}^{0} b_ip^i \right)+c_0.$$

Обозначив $c_k=a_k$ и $b_{k-1}=c_k + [a_{k-1}+c_k]_{p-1}$ получим $c_k-1 = \{a_{k-1}+ck\}_{p-1}$ и

$\sum_{i=k}^{0} a_ip^i =(p-1) \sum_{i=k-2}^{0} b_ip^i +c_0$, где $b_i(i=n-1...0)$ и c_0 определяются по формулам (1), т. е. формулы (1) верны и для $n=k$.

Для случая $a_n=p-1$ можно рассматривать представление

$$(0a_n\ldots a_0)_p,$$

а для него уже применимы формулы (1). Поэтому в общем случае имеем

$$[(a_n\ldots a_0)_p]_{\mathrm{p}-1} = (b_n\ldots b_0)_p, \quad \{(a_n\ldots a_0)_p\}_{p-1} = c_0,$$

где $\mathrm{b}_i(i = n...0)$ и c_0 определяются по рекуррентным формулам

$$b_i = c_i+1+[a_i+c_{i+1}]_{p-1}, \quad c_i = \{a_i+c_i+1\}_{p-1}, \quad c_n+1 = 0. \tag{2}$$

Заметим, что $0 \le a_i + c_{i+1} \le 2p-3$. Поэтому $0 \le [a_i + c_{i+1}]_{p-1} \le 1$ и имеем

$$[a_i+c_i+1]_{p-1} = \begin{cases} 0, & \text{если } a_i+c_i+1 < p-1, \\ 1, & \text{если } a_i+c_i+1 \ge p-1; \end{cases}$$

$$\{a_i+c_i+1\}p-1 = \begin{cases} a_i+c_i+1, & \text{если } a_i+c_i+1 < p-1 < p-1, \\ a_i+c_i+1-\text{p}+1, & \text{если } a_i+c_i+1 \ge p-1 \ge p-1, \end{cases}$$

Рассмотрим использование алгоритма деления на $p-1$ для перевода числа 3973 из десятичной системы счисления в девятеричную.

$$\begin{array}{llllll}
 & & 3 & 9 & 7 & 3 \\
\text{ci} & 0 & 3 & 3 & 1 & \boxed{4} \\
\text{bi} & & 0 & 4 & 4 & 1 \\
\hline
\text{ci} & & 0 & 4 & 8 & \boxed{0} \\
\text{bi} & & & 0 & 4 & 9 \\
\hline
\text{ci} & & & 0 & 4 & \boxed{4} \\
\text{bi} & & & & 0 & 5 \\
\hline
\text{ci} & & & & 0 & \boxed{5} \\
\text{bi} & & & & & 0
\end{array}$$

Таким образом, $(3973)_{10} = (5404)_9$.

1.2. Алгоритм деления на $p+1$. Заметим, что число $p+1$ в p-ичной системе счисления имеет вид $(11)_p$, поэтому $k(p+1)$, где k – базисное число, имеет вид $(kk)_p$.

Для двухзначного числа $(a_1a_0)_p$ имеем

$$a_1\text{p}+a_0 = a_1(p+1) + (a_0-a_1) =$$

$$= \text{a}_1(p+1)+(\text{p}+1)[a_0-a_1]_{\text{p}+1}+\{a_0-a_1\}p+1 =$$

$$= (p+1)(a_1+[a_0-a_1]_{\text{p}+1})+\{a_0-a_1\}\text{p}+1.$$

Заметим, что $-p < a_0-a_1 < p$. Поэтому $-1 \le [a_0-a_1]_{\text{p}+1} \le 0$ и

$$[a_0-a_1]_{p+1} = \begin{cases} -1, & \text{если } a_0 < a_1, \\ 0, & \text{если } a_0 \ge a_1. \end{cases}$$

При этом $[a_0-a_1]_{p+1} = -1$, если $a_1 > 0$, т.е. $0 \le a_1+[a_0-a_1]_{p+1} < p$. Отсюда имеем

$$[(a_1a_0)_p]_{p+1} = a_1+[a_0-\text{a}_1]_{p+1}, \quad \{(a_1a_0)_p\}_{p+1} = \{a_0-a_1\}_{p+1}.$$

Методом индукций по n можно показать, что

$$[(a_n\ldots a_0)_p]_{p+1} = (b_{n-1}\ldots b_0)_p, \quad \{(a_n\ldots a_0)_p\}_{p+1}=c_0,$$

где $b_i(i=n-1\ldots0)$ и c_0 определяются по следующим рекуррентным формулам:

$$b_i = c_i+1+[a_i-c_i+1]_{p+1}, \quad c_i = \{a_i-c_{i+1}\}_{p+1}, \quad c_n=a_n. \tag{3}$$

Отметим, что $-(p+1) < a_i-c_{i+1} < p$.

Поэтому $-1 \le [a_i-c_{i+1}]_{p+1} \le 0$ и имеем

$$[a_i-c_{i+1}]_{p+1} = \begin{cases} 0, & \text{если } a_i \ge c_{i+1}, \\ -1, & \text{если } a_i < c_{i+1}; \end{cases}$$

$$\{a_i-c_{i+1}\}_{p+1} = \begin{cases} a_i-c_i+1, & \text{если } a_i \ge c_{i+1}, \\ p+1+a_i-c_{i+1}, & \text{если } a_i < c_{i+1}; \end{cases}$$

Рассмотрим использование алгоритма деления на $p+1$ для перевода числа 3276 из восьмеричной системы в девятеричную.

$$\begin{array}{ccccc}
 & 3 & 9 & 7 & 6 \\
c_i & 3 & 10 & 10 & \boxed{7} \\
b_i & & 2 & 7 & 7 \\
\hline
c_i & & 2 & 5 & \boxed{2} \\
b_i & & & 2 & 5 \\
\hline
c_i & & & 2 & \boxed{3} \\
b_i & & & & 2 \\
\hline
c_i & & & & \boxed{2}
\end{array}$$

Таким образом, имеем $(3276)_8 = (2327)_9$.

1.3. Общая форма алгоритмов деления на $p-1$ и на $p+1$. Алгоритмы деления на p—1 и $p+1$ в общем виде могут быть записаны следующим образом:

$$[(a_n\ldots a_0)_p]_p\pm1 = (b_n\ldots b_0)_p, \ \{(a_n\ldots a_0)_p\}_p\pm1 = c_0,$$

где $b_i(i = n\ldots0)$ и c_0 определяются по рекуррентным формулам

$$b_i=c_{i+1}+[a_i\pm_i+1]_{p\pm1}, \ \ c_i=\{a_i\pm c_{i+1}\}p\pm1, \ \ c_{n+1}=0.$$

1.4. Другое применение алгоритмов деления на $p-1$ и на $p+1$. Одним из способов построения помехозащитных кодов является контроль по модулю, при котором вместе с каждым числом рассматривается вычет данного числа по некоторому основанию r.

В работе [3] приведены значения r для десятичной и двоичной систем и правила получения вычетов. Применение алгоритмов деления на $p-1$ и на

$p+1$ позволяет в ряде случаев упростить правила получения вычетов, а также расширить в случае двоичной системы область значений r ($r = 9$ и 17).

Рассмотрим, например, приведенное в [3] правило получения вычета по модулю 11 для десятичной системы. «Остаток от деления любого десятичного числа на 11 находится путем разбиения этого числа на пары разрядов, вычисления остатков от деления на 11 полученных таким образом двухзначных чисел и суммированием затем этих остатков по модулю 11». Заметим, что остается неясным, как получать вычет от двухзначного числа.

Правило получения вычета по модулю 11, основанное на алгоритме деления на $p+1$ с учетом того, что в этом случае нас не интересует частное от деления, может быть сформулировано следующим образом:

$$c_i = \{a_i - c_{i+1}\}_{11} \quad (i = n-1 \ldots 0),$$

где $c_n = a_n$, а c_0 есть искомый вычет.

Для сравнения приведем пример вычисления вычета по модулю 11 для числа 7 839 756 840 с помощью обоих правил.

$$\{7\,839\,756\,840\}_{11} = \{\{78\}_{11} + \{39\}_{11} + \{75\}_{11} + \{68\}_{11} +$$

$$+ \{40\}_{11}\}_{11} = \{1 + 6 + 9 + 2 + 7\}_{11} = \{25\}_{11} = 3,$$

$$7\,839\,756\,840,$$

$$c_i \; 7\,127\,051\,78\boxed{3},$$

$$\text{т. е. } \{7\,839\,756\,840)_{11} = c_0 = 3.$$

2. Алгоритмы умножения на $p-1$ и на $p+1$

Рассмотрим правила получения произведения целого числа N, имеющего представление $(a_n \ldots a_0)_p$, на p—1 и на $p+1$.

2.1. Алгоритм умножения на $p-1$. Методом индукции по n можно показать, что

$$(p-1)(a_n \ldots a_0)_p = (b_n+1 \ldots b_0)_p$$

где $b_i (i=0 \ldots n+1)$ определяются по следующим рекуррентным формулам:

$$b_i = \{c_i - a_i\}_p, \quad c_{i+1} = a_i + [c_i - a_i]_p, \quad c_0 = a_{n+1} = 0 \qquad (4)$$

При использовании этого алгоритма для перевода целых чисел $p \rightarrow p+1$ c_0 на каждом шаге полагается равным очередной p-ичной цифре числа. Покажем применение алгоритма при переводе числа 3276 из восьмеричной системы в девятеричную (этот пример рассмотрен ранее в пункте 1.2).

$$\begin{array}{lllll}
 & 3 & 2 & 7 & 6 \\
c_i & 2 & 2 & & \\
b_i & 2 & 8 & & \\
\hline
c_i & 2 & 7 & 7 & \\
b_i & 2 & 5 & 8 & \\
\hline
c_i & 2 & 5 & 7 & 6 \\
b_i & 2 & 3 & 2 & 7.
\end{array}$$

Таким образом, $(3276)_8 = (2327)_9$. Применяя алгоритм для умножения дробных чисел на p—1, т. е. при переводе $p \to p-1$, имеем

$$(p-1)(\cdot a_1 \ldots a_n) = (c_1)_{p-1} + (\cdot b_1 \ldots b_n)_p,$$

где $b_i (i = n \ldots 1)$ и c_1 определяются по рекуррентным формулам

$$b_i = \{c_i + 1 - a_i\}_p, \quad c_i = a_i + [c_i + 1 - a_i]_p, \quad c_n + 1 = 0.$$

Приведем пример использования этого алгоритма для перевода числа $\cdot 87514$ из девятеричной системы в восьмеричную.

$$\begin{array}{lllllll}
 & 8 & 7 & 5 & 1 & 4 & \\
c_i & \boxed{7} & 6 & 4 & 1 & 3 & 0 \\
b_i & 7 & 6 & 5 & 2 & 5 & \\
\hline
c_i & \boxed{6} & 5 & 4 & 2 & 4 & 0 \\
b_i & 7 & 7 & 6 & 2 & 4 & \\
\hline
c_i & \boxed{6} & 6 & 5 & 2 & 3 & 0 \\
b_i & 8 & 7 & 5 & 1 & 5 & \\
\hline
c_i & \boxed{7} & 6 & 4 & 1 & 4 & 0 \\
b_i & 7 & 6 & 5 & 3 & 4. &
\end{array}$$

Таким образом, $(\cdot 87\,514)_9 = (\cdot 7667 \ldots)_8$.

2.2. Алгоритм умножения на $p+1$. Методом индукции по n можно показать, что

$$(p+1)\{a_n \ldots a_0)_p = (b_{n+2} \ldots b_0)_p,$$

где $b_i (i = 0 \ldots n+2)$ определяются по рекуррентным формулам

$$b_i = \{c_i + a_i\}_p, \quad c_{i+1} = a_i + [c_i + a_i]_p, \quad c_0 = a_n + 1 = a_{n+2} = 0. \tag{5}$$

При использовании алгоритма для перевода целых чисел $p \to p-1$ c_0 на каждом шаге полагается равным очередной p-ичной цифре числа. Покажем применение этого алгоритма для перевода числа 3973 из десятичной системы в девятеричную.

$$
\begin{array}{lccccc}
 & 0 & 3 & 9 & 7 & 3 \\
c_i & 0 & 3 & & & \\
b_i & 0 & 3 & & & \\
\hline
c_i & 0 & 4 & 9 & & \\
b_i & 0 & 4 & 3 & & \\
\hline
c_i & 0 & 4 & 4 & 7 & \\
b_i & 0 & 4 & 8 & 1 & \\
\hline
c_i & 0 & 5 & 9 & 1 & 3 \\
b_i & 0 & 5 & 4 & 0 & 4.
\end{array}
$$

Таким образом, $(3973)_{10} = (5404)_9$.

Применяя этот алгоритм умножения для вычисления произведения дробного числа на $p+1$, т. е. при переводе $p \to p+1$, имеем

$$(p+1)\{\cdot a_1 \ldots a_n)_p = (c_1)_{p+1} + (\cdot b_1 \ldots b_n)p,$$

где $b_i (i = n \ldots 1)$ и c_1 определяются по рекуррентным формулам

$$b_i = \{c_i + 1 + a_i\}_p, \quad c_i = a_i + [c_i + 1 + a_i]_p, \quad c_n + 1 = 0.$$

Приведем пример использования этого алгоритма для перевода числа ·1 101 011 из двоичной системы в троичную.

$$
\begin{array}{lcccccccc}
 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & \\
c_i & \boxed{10} & 10 & 1 & 10 & 1 & 10 & 1 & 0 \\
b_i & 1 & 0 & 0 & 0 & 0 & 0 & 1 & \\
\hline
c_i & \boxed{1} & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
b_i & 1 & 0 & 0 & 0 & 0 & 1 & 1 & \\
\hline
c_i & \boxed{1} & 0 & 0 & 0 & 1 & 10 & 1 & 0 \\
b_i & 1 & 0 & 0 & 1 & 0 & 0 & 1 & \\
\hline
c_i & \boxed{1} & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
b_i & 1 & 0 & 1 & 1 & 0 & 1 & 1 &
\end{array}
$$

Таким образом, $(\cdot 1\,101\,011\,)_2 = (\cdot 2111\ \ldots)_3$.

2.3. Общая форма алгоритмов умножения на $p-1$ и на $p+1$. Алгоритмы умножения целых чисел на $p-1$ и на $p+1$ в общем виде могут быть записаны следующим образом:

$$(p \pm 1)(a_n \ldots a_0)_p = (b_n + 2 \ldots b_0)p,$$

где $b_i (i=0 \ldots n+2)$ определяются по рекуррентным формулам

$$b_i = \{c_i \pm a_i\}_p, \quad c_i + 1 = a_i + [c_i \pm a_i]p, \quad c_0 = a_n \pm 1 = a_n + 2 = 0.$$

3. Заключение

Рассмотренные алгоритмы деления и умножения целых чисел на $p-1$ и на $p+1$ в p-ичной системе счисления сводятся к операциям над разрядами (сложению, вычитанию и вычислению вычетов). Использование этих алгоритмов позволяет сформулировать простые правила переводов $p \to p-1$ и $p \to p+1$. Простота этих правил позволяет рассчитывать на то, что их аппаратная реализация не будет связана с затратой значительного количества оборудования или времени.

Очевидно, что эти алгоритмы не решают в общем случае проблемы перевода из одной системы счисления в другую. Однако в области их применимости имеются следующие пары чисел:

(9, 10), (2, 3), (3, 4), (8, 9),

т. е. с помощью этих алгоритмов разрешимы вопросы переводов из троичной системы счисления в десятичную и в наиболее распространенную в настоящее время в вычислительной технике двоичную системы, а также обратные переводы (10→3 и 2→3).

ЛИТЕРАТУРА

1. Брусенцов Н.П. Об использовании троичного кода и трехзначной логики в цифровых машинах. В сб.: «Вычислительная техника и вопросы кибернетики», вып. 7. Изд-во МГУ, 1970.

2. Brins T. Data transmission is faster with ternary coding. «Electronics», 47, № II, 119—120, 1974.

3. Карцев М.А. Арифметика цифровых машин. М, «Наука», 1969.