Asian Journal of Information Technology 6 (10): 1071-1073, 2007
ISSN: 1682-3915

The Conversion of Numbers from Positional Notation with One Radix into Positional Notation with Bigger Radix

Afif Mghawish
Department of Computer Science, Al-Zaytoonah University, Jordan

Abstract: The number conversion algorithm from positional notation with one radix into positional notation with bigger radix has been presented in this study. The algorithm correctness has been proved. Suggested algorithm is the most efficient among known ones because the conversion is carried out in m-1 steps where m is the number of digits. The number of steps can vary from m-1 up to m (m-1)/2, and calculation at each step is limited to multiplication of two one-digit numbers and subtraction (addition) of the product from the result received at the previous step (in case of a variant of algorithm with number of steps m (m-1)/2).

Key words: Number notation, positional number notation, radix, radix conversion algorithm

INTRODUCTION

Number conversion algorithms from one positional system into another have been known since long time ago (Iguchi *et al*., 2007; Donald, 1998; Viktorov *et al*., 1977; Plauger, 1992). To find a new number conversion algorithm nowadays is an impossible event. The number conversion from positional notation with radix 8 into decimal has been presented in Croy (1961). Hence this algorithm has not been modified for conversion of numbers from positional notation with one arbitrary radix S into positional notation with bigger arbitrary radix R. Two radix conversion algorithms have been presented: the first one is used for integers and another one is used for fractions.

THE MAIN RESULTS

We are given a number $(A)_S = a_m S^m + a_{m-1} S^{m-1} + \ldots + a_1 S^1 + a_0$, which it is necessary to convert into positional notation with radix R, i.e. to find $(A)_R$. Let us assume, that $R > S$. The number $(A)_S$ may be interpreted as another number $(B)_R = a_m R^m + a_{m-1} R^{m-1} + \ldots + a_1 R^1 + a_0$, that is presented in positional notation with radix R, which digits a_i ($i = 1, 2, \ldots, m$) are the same as corresponding digits of $(A)_S$. Therefore, the difference between $(B)_R$ and $(A)_S$ is $(B)_R - (A)_S = a_m(R^m - S^m) + a_{m-1}(R^{m-1} - S^{m-1}) + \ldots + a_1(R^1 - S^1)$. Evidently, that the number $(A)_S$, can be always interpreted as the number $(B)_R$ with radix R, because its digits a_i do not exceed $R - 1$.

The algorithm that converts the integer A from positional notation with radix S into positional notation with radix R is presented below:

Algorithm 1: Integer numbers conversion.

1. Find

$$(B)_R - a_m(R-S)R^{m-1} = a_m^{(1)}R^m + a_{m-1}^{(1)}R^{m-1} + a_{m-1}^{(1)}R^{m-2} + \ldots + a_1R^1 + a_0$$

2. Find

$$(B)_R^{(1)} - (a_m^{(1)}R + a_{m-1}^{(1)})(R-S)R^{m-2} = a_m^{(2)}R^m + a_{m-1}^{(2)}R^{m-1} + a_{m-2}R^{m-2} + \ldots + a_1R^1 + a_0$$

.
.
.

m. Find

$$(B)_R^{(m-1)} - (a_m^{(m-1)}R^m + a_{m-1}^{(m-1)}R^{m-1} + a_{m-2}^{(m-1)}R^{m-2} + \ldots + a_1^{(1)})(R-S) =$$
$$= a_m^{(m)}R^m + a_{m-1}^{(m)}R^{m-1} + a_{m-2}^{(m-1)}R^{m-2} + \ldots + a_1^{(2)}R^1 + a_0^{(1)}$$

Remark: The digit upper index value (presented in parenthesis) corresponds to the number of modifications that have been done during algorithm execution.

Let us prove that $(B)_R^{(m)} = (A)_R$

Proof: The following numbers have been subtracted from $(B)_R$ during algorithm execution:
At the first step: $a_m(R-S)R^{m-1}$;
At the second step: $(a_{m-1} + a_m S)(R-S)R^{m-2}$

.
.
.

At the ith step: $(a_{m-i} + a_{m-i+1} S + \ldots + a_m S^i)(R-S)R^{m-i+1}$

.
.
.

At the mth step: $(a_1 + a_2 S^1 + \ldots + a_m S^{m-1})(R-S)$

Let us find the sum of the numbers that have been subtracted at m steps

$(a_m R^{m-1} + (a_{m-1} + a_m S)R^{m-2} + \ldots + (a_1 + a_2 S^1 + \ldots + a_m S^{m-1}))(R - S) =$
$= a_m(R^m - S^m) + a_{m-1}(R^{m-1} - S^{m-1}) + \ldots + a_1(R - S) = B - A$

Thus, when the number $(A)_S$ is interpreted as $(B)_R$ the value of number is increased by B–A and then during algorithm execution this difference is subtracted from $(B)_R$.
Example 1. Convert the number $(1423)_5$ into numeric notation with radix 8.

$$S = 5, R = 8, (R - S) = 3$$

Note that all operations have been carried out in radix-8 arithmetic.

```
1423
–
   3      1×3=3
1123
–
  33      11×3=33
0573
–
 215      57×3=215
 356
```

$(1423)_5 = (356)_8$

It is clear that any fraction may be presented as follows $(A)_S = a_{-1} S^{-1} + a_{-2} S^{-2} + \ldots + + a_{-m} S^{-m}$. To convert the number $(A)_S$ into the positional notation with radix R, it is necessary to interpret given number as another number $(F)_R = a_{-1} R^{-1} + a_{-2} R^{-2} + \ldots + a_{-m} R^{-m}$, presented in positional notation with radix R, which digits a_i (i = 1, 2, . . . , m) are the same as corresponding digits of $(A)_S$. Therefore, the difference between $(A)_S$ and $(F)_R$ is

$$(A)_S - (F)_R = a_1 (S^{-1} - R^{-1}) + \ldots + a_{m-1} (S^{-m+1} - R^{-m+1}) + a_m (S^{-m} - R^{-m})$$

Algorithm 2: Fraction number conversion

1. Find:
$(F)_R + a_{-m}(S^{-1} - R^{-1})R^{-m+1} = a_{-1}^{(1)} R^{-1} + a_{-2}^{(1)} R^{-2} + \ldots + a_{-m+1}^{(1)} R^{-m+1} + a_{-m}^{(1)} R^{-m}$

2. Find:
$(F)_R^{(1)} + (a_{-m+1}^{(1)} + a_{m-1}^{(1)} R^{-1}) (S^{-1} - R^{-1})R^{-m+2} =$
$a_{-1}^{(2)} R^{-1} a_{-2}^{(2)} R + \ldots + a_{-m+2}^{(2)} R^{-m+2} a_{-m+1}^{(2)} R^{-m+1} + a_{-m}^{(2)} R^{-m}$

.
.
.

m. Find:
$(F)_R^{(m-1)} + (a_{-1}^{(m-1)} + a_{-2}^{(1)} R^{-1} + \ldots + a_{-m}^{(m-1)} R^{-m+1}) (S^{-1} - R^{-1}) =$
$a_{-1}^{(m)} R^{-1} a_{-2}^{(m)} R^{-2} + \ldots + a_{-m}^{(m)} R^{-m} = (F)_R^{(m)}$

Remark: The digit upper index value (presented in parenthesis) corresponds to the number of modifications that have been done during algorithm execution.

Let us prove that $(F)_R^{(m)} = (A)_R$

Proof:

Let us prove that $(F)_R^{(m)} = (A)_R$

The following numbers have been added to $(F)_R$ during algorithm execution:

At the first step : $a_{-m}(S^{-1} - R^{-1})R^{-m+1}$;
At the second step : $((a_{-m+1} + a_m R^{-1}) + a_{-m} (S^{-1} - R^{-1}))(S^{-1} - R^{-1}) R^{-m+2} =$
$= (a_{-m+1} + a_{-m} S^{-1}) (S^{-1} - R^{-1})R^{-m+2}$;

.
.
.

At the ith step : $(a_{-m+i-1} + a_{-m+i-2} S^{-1} + \ldots + a_{-m} S^{-i+1})(S^{-1} - R^{-1})R^{-m+i}$;

.
.
.

At the mth step : $(a_{-1} + a_{-2} S^{-1} + \ldots + a_{-m} S^{-m+1}) (S^{-1} - R^{-1}) R^0$

Let us find the sum of the numbers that have been added to $(F)_R$ at m steps:

$a_{-m}(S^{-1} - R^{-1})R^{-m+1} + (a_{-m+1} + a_{-m} S^{-1}) (S^{-1} - R^{-1})R^{-m+2} + \ldots$
$+ (a_{-m+i-1} + a_{-m+i-2} S^{-1} + \ldots + a_{-m} S^{-i+1})(S^{-1} - R^{-1}) R^{-m+i} + \ldots$
$+ (a_{-1} + a_{-2} S^{-1} + \ldots + a_{-m} S^{-m+1})(S^{-1} - R^{-1})R^0 =$
$= a_{-1}(R^{-1} - S^{-1}) + a_{-2}(R^{-2} - S^{-2}) + \ldots + a_{-m+1}(R^{-m+1} - S^{-m+1})$
$+ a_{-m}(R^{-m} - S^{-m}) = A - F$

Thus, when the number $(A)_S$ is interpreted as $(F)_R$ the value of number is decreased by A-F, and then during algorithm execution this difference is added to $(F)_R$.

Example 2: Convert the number $(0.0111)_2$ into numerical system with radix 10.

$$S = 2, R = 10, S^{-1} - R^{-1} = (1/2 - 1/10) = 4/10$$

Note that all operations have been carried out in decimal arithmetic. Digits are multiplied by numerator 4.

$$
\begin{array}{ll}
0.0111 & \\
+ & \\
\underline{0.0004} & 0.0001\times 4 = 0.0004 \\
0.0115 & \\
+ & \\
\underline{0.0060} & 0.0015\times 4 = 0.0060 \\
0.0175 & \\
+ & \\
\underline{0.0700} & 0.0175\times 4 = 0.0700 \\
0.0875 & \\
+ & \\
\underline{0.3500} & 0.0875\times 4 = 0.3500 \\
0.4375 & \\
\end{array}
$$

$(0.0111)_2 = (0.4375)_{10}$

It is necessary to note, that if the common multiple of S and R is R, then the proper digits of the number are multiplied by the numerator of the difference (1/S-1/R) = (a/R-1/R) = (a-1)/R. Otherwise instead of the difference (1/S-1/R) it is necessary to use the nearest fraction b/R, that will reduce the accuracy of conversion. However, if b can be fractional it is possible to provide any required accuracy.

CONCLUSION

Suggested algorithm is the most efficient among known ones because the conversion is carried out for m-1 steps where m is the number of digits in the converted number. The number of steps can vary from m-1 up to m (m-1)/2 and calculation on each step is limited to multiplication of two one-digit numbers and subtraction (addition) of the product from the result received on the previous step (in case of a variant of algorithm with number of steps m(m-1)/2).

REFERENCES

Croy, J.E., 1961. Rapid Technique of Manual of Machine Binary-to-Decimal Integer Conversion Using Decimal Radix Arithmetic, IRE. Trans. Elec. Comput., EC-10, 4: 777.

Donald E. Knuth, 1998. The Art of Computer Programming, Seminumerical Algorithms. 3rd Edn. Addison-Wesley, Vol. 2.

Iguchi, Y., T. Sasao and M. Matsuura, 2007. Design Methods of Radix Converters Using Arithmetic Decomposition IEICE. Trans. Inform. Sys., E90-D, 6: 905-914.

Plauger, P.J., 1992. The Standard C Library, Prentice-Hall.

Viktorov, O.V., V.A. Ostafin, A.M. Romankevich and A.I. Yatsunov, 1977. Code Converter, Device Invention Author's Certificate[1] 581469 USSR, Invention Author's Certificate Bull., pp: 43.