Alternative Algorithms for Integer Base Conversion

Author: Steve Palmer (remlaps@ccil.org)

Date: February 19, 2007

(Rough) DRAFT **DRAFT**

Table of Contents

0.) Revision History

Date	Revision
2/19/7	Initial draft created
2/20/7	Additional clarification added to Introduction (section 1.).
2/22/7	Formatting

1.) Introduction

This document describes some observed methods for converting integers from one base of representation to another. These methods may be unusual, in that they operate digit-wise on the integer. Using these methods, it is never necessary to divide from, or even to calculate the actual value of the represented integer; only the values of the digits are needed.

The algorithms presented provide the capability to convert base of representation when certain conditions are met. The first algorithm, from section 2, converts bases using an offset, e. Using this algorithm, one can convert from base B to base B ± e. The second algorithm, in sections 3 and 4, enables conversion when the source base is an even multiple of the target (section 3) or when the target base is an even multiple of the source (section 4). All of the conversion algorithms presented here also depend on a technique which has been called digit normalization[1].

It is unlikely that these algorithms are new, but I haven't found them described anywhere. I am presently unaware of any rigorous proof having been done for these algorithms. I have proved them, informally and to my own satisfaction, out to 8 digits.

2.) Converting Between Bases by an Offset

This section describes an algorithm which will convert a numeral from one base to another, using the offset between bases. The same algorithm can also convert a polynomial in terms of X to an equivalent polynomial in terms of [X - offset].

This algorithm depends on matrix multiplication by Pascal's Triangle in a particular orientation. Pascal's triangle has been used for base conversion[2], but I haven't found anyone who did it explicitly through matrix multiplication.

2.1.) Preconditions:

- A positive integer numeral in some integer base, or a polynomial in terms of X.
- The target base or a desired offset between bases or term of polynomial.

2.2.) The Matrix Construction

Pascal's Triangle should be oriented in a matrix as shown in these examples.

1 digit - $\begin{matrix} 1 \end{matrix}$

2 digits - $\begin{matrix} 1 & 1 \\ 0 & 1 \end{matrix}$

3 digits - $\begin{matrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{matrix}$

4 digits - $\begin{matrix} 1 & 3 & 3 & 1 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{matrix}$

5 digits - $\begin{matrix} 1 & 4 & 6 & 4 & 1 \\ 0 & 1 & 3 & 3 & 1 \\ 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{matrix}$

etc...

2.3.) The steps:

I. If given just the source and target base, calculate the offset, e, by subtracting the target base from the source base. If given the source base and an offset, calculate target base by subtracting offset from source base.
II. Start with a numeral in a base B number system or a polynomial in terms of X.
III. Construct a 1 row array using base 10, ordinal, values of digits or coefficients from I. Call it N.
IV. Construct a square array, using a right-rooted Pascal's Triangle. The number of rows and columns in the square should be the same as the number of elements in the array from II. Call it P.
V. Raise P to e.
VI. Matrix Multiply $R=N \cdot P^e$.

R now contains coefficients for a polynomial in terms of (X - Offset).

To convert R into a numeral in base [X-Offset], use digit normalization as described in section 5.

3.) Converting Between Bases When the Source Base is a Mutliple of the Target Base

3.1.) Preconditions

- A positive integer in some base, or a polynomial in terms of X.
- The target base or term of polynomial or else a desired reduction factor between bases
- Source base must be an integer multiple of target base. Call Source/Target F.

3.2.) The Steps

I. Split the numeral into digits or create a list of coefficients from the polynomial.
II. Put powers of F into a list, increasing in order from right to left.
III. Multiply digits, from I, by corresponding powers of F from II.
IV. If necessary, normalize digits into target base by "carrying", as described in section 5.

4.) Converting Between Bases When the Target Base is a Multiple of the Source Base.

4.1.) Preconditions

- A positive integer in some base, or a polynomial in terms of X.
- The target base or else a desired multiplier to be multiplied by the source base.
- Target base must be an integer multiple of source base. Call Target/Source M.

4.2.) The Steps

I. Make a list from the digits (or coefficients).
II. Put powers of M into a list, increasing from left to right.
III. Multiply digits from I. by corresponding powers of M, from II.
IV. Divide each multiplication result by the highest power of M.
   1. Divide from left to right.
   2. For each place, multiply remainder by base and shift it rightwards.
V. If necessary, normalize as described in section 5.

5.) Digit Normalization

If the goal is base conversion, an additional step is usually needed to convert coefficients into legal digits for a numeral in a base. Normalization is done with standard carrying/borrowing as follows.

```
Let coefficients = {* dk, dk-1, dk-2, ..., d2, d1, d0 }
For each digit, di from right to left, do:
  While di < 0
    di <- di + Target_Base
    di+1 <- di+1 -1
  End
  While di >= Target_Base
    di <- di - Target_Base
    di+1 <- di+1 + 1
  End
End
```

*Insert or delete leading 0s as is useful.
Note: A variant of normalization also exists for negative target bases.
Note: This can also be done using div/mod functions rather than looping.

6.) Examples

6.1.) Pascal's Triangle Examples - Base conversion by an offset

6.1.1.) Example 1: 152_{10} to base 9

1. If given just the source and target base, calculate the offset, e, by subtracting the source base from the target base. If given the source base and an offset, calculate target base by adding offset to source base.

   Source = 10, Target = 9, Offset = Source - Target = 1.

2. Start with a numeral in a base B number system or a polynomial in terms of X.

   Start with 152_{10}

3. Construct a 1 row array using digits or coefficients. Call it N.

   N = [1 5 2]

4. Construct a square array, using a right-rooted Pascal's Triangle. The number of rows and columns in the square should be the same as the number of elements in N. Call it P.

   $$P = \begin{matrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{matrix}$$

5. Raise P to e. Getting P^e.

   $P^e = P$

6. Matrix Multiply $R=N P^e$.

   $$R = [1\ 5\ 2] \begin{matrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{matrix} =[(1) \quad (2 + 5) \quad (1 + 5 + 2)] = [1\ 7\ 8]$$

So 152_{10} goes to 178_9.
Verification: $1(9^2) + 7(9) + 8 = 81 + 63 + 8 = 152_{10}$

6.1.2.) Example 2: 189_{10} to base 11.

1. If given just the source and target base, calculate the offset, e, by subtracting the source base from the target base. If given the source base and an offset, calculate target base by adding offset to source base.

   Source base = 10, Target base = 11, Offset = Source - Target = -1

2. Start with a numeral in a base B number system or a polynomial in terms of X.

   Numeral = 189 (polynomial = $X^2 + 8X + 9$)

3. Construct a 1 row array using digits or coefficients from I. Call it N.

   N=[1 8 9]

4. Construct a square array, using a right-rooted Pascal's Triangle. The number of rows and columns in the square should be the same as the number of elements in N. Call it P.

$$P = \begin{matrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{matrix}$$

5. Raise P to e. Getting P^e.

$$P^{-1} = \begin{matrix} 1 & -2 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{matrix}$$

6. Matrix Multiply R=N P^e.

$$R = [1\ 8\ 9] \begin{matrix} 1 & -2 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{matrix} = [(1)\ (-2 + 8)\ (1 -8 + 9)] = [1\ 6\ 2]$$

So 189_{10} goes to 162_{11} [or $X^2 + 8X + 9$ goes to $1(X+1)^2 + 6(X+1) + 2$].

Verification: 1(121) + 6(11) + 2 = 121 + 66 + 2 = 189.

6.1.3.) Example 3: 173_{10} to base 4, includes Digit Normalization

$$N = [1\ 7\ 3],\ P = \begin{matrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{matrix},\ P^6 = \begin{matrix} 1 & 12 & 36 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{matrix}$$

$$NP^6 = [1\ 7\ 3] \begin{matrix} 1 & 12 & 36 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{matrix} = [(1)\quad (12 + 7)\quad (36 + 42 + 3)] = [1\ 19\ 81]$$

Digit Normalization (base 4)

d_3	d_2	d_1	d_0
0	1	19	81
0	1	20	77
0	1	21	73
0	1	22	69
0	1	23	65
0	1	24	61
0	1	25	57
0	1	26	53
0	1	27	49
0	1	28	45
0	1	29	41
0	1	30	37
0	1	31	33
0	1	32	29
0	1	33	25
0	1	34	21
0	1	35	17
0	1	36	13
0	1	37	9
0	1	38	5
0	1	39	1

d_3	d_2	d_1	d_0
0	2	35	1
0	3	31	1
0	4	27	1
0	5	23	1
0	6	19	1
0	7	15	1
0	8	11	1
0	9	7	1
0	10	3	1
1	6	3	1
2	2	3	1

So 173_{10} goes to 2231_4

[and $X^2 + 7X + 3$ goes to $\{1(X-6)^2 + 19(X-6) + 81$ and $2(X-6)^3 + 2(X-6)^2 + 3(X-6) + 1]$

Verification: $2231_4 = 2(64) + 2(16) + 3(4) + 1 = 128 + 32 + 12 + 1 = 173_{10}$

6.2.) Multiples of Bases Examples

6.2.1.) Example 4: 1776 from base 10 to base 2.

0.) Start with: 1776_{10}
1.) Split into digits: {1,7,7,6}
2.) Get increasing powers of 10/2 = 5 (right to left): {125,25,5,1}
3.) Multiply digits by corresponding powers of 5: {125,175,35,6}
4.) Normalize into base 2 by "carrying".

```
{125,175,35,6}
-> {125, 175, 38, 0}                  # 38 = 35 + 3
-> {125, 194, 0, 0}                   # 194 = 175 + 19
-> {222, 0, 0, 0}                     # 222 = 125 + 97
-> {111, 0, 0, 0, 0}                  # 111 is a new place
-> {55, 1, 0, 0, 0, 0}                # 55 is a new place
-> {27, 1, 1, 0, 0, 0, 0}             # 27 is a new place
-> {13, 1, 1, 1, 0, 0, 0, 0}          # ...
-> {6, 1, 1, 1, 1, 0, 0, 0, 0}        # ...
-> {3, 0, 1, 1, 1, 1, 0, 0, 0, 0}     # ...
-> {1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0}  # ...
```

So $1776_{10} = 11011110000_2$

6.2.2.) Example 5: 1011 from base 2 to base 10.

0.) Start with: 1011
1.) Split into digits: {1,0,1,1}
2.) Get increasing powers of 10/2 = 5 (left to right): {1,5,25,125}
3.) Multiply digits by corresponding powers of 5: {1,0,25,125}
4.) Divide each place by 125, borrow remainder:

```
{1,0,25,125}
-> {0,10,25,125}     # 10 = 10 * 1 + 0
-> {0,0,125,125}     # 125 = 10 * 10 + 25
-> {0,0,1,125}       # 1 = 125/125
-> {0,0,1,1}         # 1 = 125/125
```

5.) Digit normalization(not needed for this example).
So $1011_2 = 11_{10}$.

7.) References

1. Harmon T. Gladwin, Communications of the ACM, 1964
2. US Patent number 40312015, Lonnie Machen, 1979
3. http://taz.cs.wcupa.edu/~spalmer/PTIConvert/matcon.cpp.txt
4. http://home.ccil.org/~remlaps/src/normalize.java.txt