A Base Conversion Algorithm

Author: Steve Palmer (remlaps@ccil.org)  Date: November, 2006

This memo describes an algorithm which can be used to convert a numeral from one base to another, using the offset between bases or to convert a polynomial in terms of X to a polynomial in terms of X - offset.

Preconditions:

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

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. Call it N.

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-bottom-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. The lower left triangle will be filled with 0's. 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).

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 = \{ d_k, d_{k-1}, d_{k-2}, ..., d_2, d_1, d_0 \}

For each digit, d_i from right to left, do:

While d_i < 0
   d_i ← d_i + Target_Base
   d_{i+1} ← d_{i+1} - 1
End

While d_i >= Target_Base
   d_i ← d_i - Target_Base
   d_{i+1} ← d_{i+1} + 1
End

*Insert or delete leading 0s as is useful.

Then, concatenate the digits into a numeral in base B-offset representation.
Array Construction

Pascal's Triangle is oriented in the matrix as shown in these examples.

1 digit -  
1

2 digits -  
1 1  
0 1  
1 2 1

3 digits -  
0 1 1  
0 0 1  
0 0 0 1

4 digits -  
1 3 3 1  
0 1 2 1  
0 0 1 1  
0 0 0 1

5 digits -  
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

6 digits -  
1 5 10 10 5 1  
0 1 4 6 4 1  
0 0 1 3 3 1  
0 0 0 1 2 1  
0 0 0 0 1 1  
0 0 0 0 0 1

eetc....

Yet to be done:

- I have proved this to 8 digits. A general proof is still needed (unless it's known).
- Analyze algorithm run time and compare to other base conversion methods (especially in the context of parallel programming).
- Optimize algorithm.
- Extend algorithm for floating point numerals.
- Extend algorithm for negative numbers.
- Extend algorithm for negative bases.
- It is probably obvious that this algorithm can be applied almost immediately to a list of numerals or polynomials. The polynomials can be placed in consecutive rows of array N and the corresponding rows in the multiplication result will contain the converted polynomials. Each row must have digits normalized if base conversion is desired.
- The multiple numeral algorithm also needs to be analyzed and optimized.
Examples:

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 $P$.
   \[
   P = \begin{bmatrix}
   1 & 2 & 1 \\
   0 & 1 & 1 \\
   0 & 0 & 1
   \end{bmatrix}
   \]

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

6. Matrix Multiply $R = N \times P^e$.
   \[
   R = \begin{bmatrix}
   1 & 2 & 1 \\
   1 & 5 & 2 \\
   0 & 1 & 1
   \end{bmatrix}
   \begin{bmatrix}
   1 & 2 & 1 \\
   0 & 1 & 1 \\
   0 & 0 & 1
   \end{bmatrix}
   = \begin{bmatrix}
   1 & 7 & 8 \\
   0 & 0 & 1
   \end{bmatrix}
   \]
   So $152_{10}$ goes to $178_9$. [or $X^2 + 5X + 2$ goes to $1(X-1)^2 + 7(X-1) + 8$]
   Verification: $1(9^2) + 7(9) + 8 = 81 + 63 + 8 = 152_{10}$
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.  
   
   \[
   \begin{array}{ccc}
   1 & 2 & 1 \\
   0 & 1 & 1 \\
   0 & 0 & 1 \\
   \end{array}
   \]

5. Raise P to e.  Getting $P^e$.  
   
   \[
   \begin{array}{ccc}
   1 & -2 & 1 \\
   0 & 1 & -1 \\
   0 & 0 & 1 \\
   \end{array}
   \]

6. Matrix Multiply $R=N \cdot P^e$.  
   
   \[
   R = [1 8 9] \begin{array}{ccc} 1 & -2 & 1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{array} = [(1) (-2 + 8) (1 -8 + 9)] = [1 6 2]
   \]

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

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

Example 3: $173_{10}$ to base 4

$N = [1 \ 7 \ 3]$, $P = [0 \ 1 \ 1]$, $P^6 = [0 \ 1 \ 6]$

$NP^6 = [1 \ 7 \ 3]$ $[0 \ 1 \ 6] = [(1 \ (12 + 7)) (36 + 42 + 3)] = [1 \ 19 \ 81]$

$0 \ 0 \ 1$

**Digit Normalization (base 4)**

<table>
<thead>
<tr>
<th>$d_3$</th>
<th>$d_2$</th>
<th>$d_1$</th>
<th>$d_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>19</td>
<td>81</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>20</td>
<td>77</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>21</td>
<td>73</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>22</td>
<td>69</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>23</td>
<td>65</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>24</td>
<td>61</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>25</td>
<td>57</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>26</td>
<td>53</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>27</td>
<td>49</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>28</td>
<td>45</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>29</td>
<td>41</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>30</td>
<td>37</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>31</td>
<td>33</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>32</td>
<td>29</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>33</td>
<td>25</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>34</td>
<td>21</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>35</td>
<td>17</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>36</td>
<td>13</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>37</td>
<td>9</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>38</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>39</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>35</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>31</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>27</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>5</td>
<td>23</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>6</td>
<td>19</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>15</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>8</td>
<td>11</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>9</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>10</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

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