

Non-Recursive Base Conversion Using a Deterministic Markov Process

Louis M. Houston

Department of Innovation and Research, The University of Louisiana at Lafayette, Lafayette, LA, USA

Email: louis.houston@louisiana.edu

How to cite this paper: Houston, L.M. (2024) Non-Recursive Base Conversion Using a Deterministic Markov Process. *Journal of Applied Mathematics and Physics*, 12, 2112-2118. <https://doi.org/10.4236/jamp.2024.126129>

Received: February 27, 2024

Accepted: June 18, 2024

Published: June 21, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

We prove that non-recursive base conversion can always be implemented by using a deterministic Markov process. Our paper discusses the pros and cons of recursive and non-recursive methods, in general. And we include a comparison between non-recursion and a deterministic Markov process, proving that the Markov process is twice as efficient.

Keywords

Base Conversion, Recursion, Euclidean Division, Geometric Series, Markov Process

1. Introduction

Mathematically, the base or radix, b is the building block of a number. A digit of a number is maximally $b - 1$. For example, the largest digit in base ten is nine. The subject matter of this paper is about converting a number from one base system to another base system and revealing the coupling between base conversion and a Markov process.

To discuss the conditions that couple non-recursive base conversion to a deterministic Markov process, we need to break down the concepts involved:

- **Non-Recursive Base Conversion:** This refers to the process of converting a number from one base to another without using recursion. The conversion process involves a series of steps that can be explicitly listed and executed sequentially, such as dividing the number by the new base and taking the remainder to form the digits of the new base.
- **Deterministic Markov Process:** A Markov process is a stochastic process where the probability of transitioning to any future state depends only on the current state and not on the sequence of events that preceded it. A determi-

nistic Markov process implies that the transitions between states are determined by a fixed rule without any randomness. Essentially, the deterministic Markov process presented in this paper has transition probabilities equal to 1.

The theorem presented in this paper is not to be confused with the Base Conversion Theorem by Matula [1]. The Base Conversion Theorem stipulates the conditions under which base conversions are one-to-one and/or onto. The theorem presented in this paper provides a specific method to produce non-recursive base conversion for positive integers and decimal numbers that are consistent with deterministic Markov processes.

1.1. The Markov Base Conversion Theorem

The base conversion of a number has historically been a recursive process that depends on the dividends and remainders from Euclidean division or the multiplication of real numbers [2] [3] [4]. We derive a process that easily converts a number from base ten to another base without recursion. The theorem is fundamentally based on the intuitive process of isolating digits in a base ten or decimal number that are separated by powers of ten and therefore are distinctly represented in the number. The process that separates the base ten elements from the number simply isolates the digits of the decimal number. The derived process does not depend on the remainders from division. By generating the process to convert between decimal and an arbitrary base, we derive the theorem of deterministic Markov base conversion. The theorem specifies base conversion between an N-digit positive decimal number less than one and any other base. While base conversion readily extends to negative and floating-point numbers [5] [6] [7] [8], we will not discuss that extension in this paper. An essential component of the theorem is the use of the floor function and the geometric series [9] [10]. We also show that a deterministic Markov base conversion is twice as efficient, mathematically, as typical base conversion methods, such as iterative or non-recursive methods.

1.2. A Markov Chain or Markov Process

A Markov chain or Markov process is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event. Informally, this may be thought of as, “What happens next depends only on the state of affairs *now*.”

A deterministic Markov process is generally non-recursive as it typically involves state transitions governed by deterministic rules or functions, which can be implemented iteratively without the need for recursive calls.

Recursive base conversion is a clear and concise method for converting numbers between bases, though it may not always be the most efficient. For small to moderately sized numbers, it provides a readable and maintainable solution. For larger numbers, an iterative approach might be more suitable to avoid potential

performance and memory issues.

Coupling non-recursive base conversion to a deterministic Markov process involves defining states that represent each stage of the conversion, establishing deterministic transition rules that depend solely on the current state, and ensuring the process terminates in a final state representing the converted number. This approach leverages the deterministic nature of both the base conversion algorithm and the Markov process framework to achieve the desired coupling.

- A deterministic Markov process describes state transitions where the next state is determined solely by the current state.
- It can be implemented either iteratively or recursively. The choice of implementation depends on the specific requirements and constraints of the problem at hand.
- An iterative implementation is generally more memory-efficient and straightforward, while a recursive implementation might be more intuitive for certain problems but risks stack overflow for large numbers of state transitions.

2. Recursive Base Conversion

We want to consider two cases: 1) base conversion for a positive integer and 2) base conversion for a positive number less than one. In case one, we want to show that a recursive process that retains the remainders from Euclidean division achieves a conversion between base ten and any other base.

Consider a positive N -digit base-ten integer x . We can write it as

$$x = \sum_{i=1}^N a_i b^{i-1}, \quad (1)$$

where $\{a_i\}$ are the digits of x in the base $b > 1$. Observe that if you divide $x = x^{(1)}$ by b , by Euclidean division you get

$$x^{(1)} = r_1 + bx^{(2)} \quad (2)$$

For some positive integers $x^{(2)} < x^{(1)}$ and $r_1 < b$. You can then recursively divide $x^{(2)}$ by b (note that this will require only a finite number of steps), obtaining

$$x = r_1 + b(r_2 + b(\dots + b(r_N))) \quad (3)$$

with $0 \leq r_i < b$. Therefore, $a_i = r_i$ as required.

Now let us examine case two. Consider a positive N -digit decimal number less than one. We can write it as

$$x = \sum_{i=1}^N a_i b^{i-1-N}. \quad (4)$$

where $\{a_i\}$ are the digits of x in the base $b > 1$. Observe that if you multiply $x = x^{(1)}$ by b , you get

$$bx^{(1)} = x^{(2)} + q_N. \quad (5)$$

and if you multiply $x = x^{(2)}$ by b you get

$$bx^{(2)} = x^{(3)} + q_{N-1}, \quad (6)$$

where q_N and q_{N-1} are positive integers. If we multiply by b^{-1} , we get

$$x^{(1)} = b^{-1}x^{(2)} + b^{-1}q_N \tag{7}$$

We can then recursively multiply by b^{-1}

$$x^{(1)} = b^{-1} \left(b^{-1}x^{(3)} + \dots + b^{-1}q_{N-1} \right) + b^{-1}q_N$$

$$x^{(1)} = b^{-1} \left(b^{-1} \left(b^{-1}x^{(4)} + b^{-1}q_{N-2} \right) + b^{-1}q_{N-1} \right) + b^{-1}q_N \quad \{N \text{ operations}\}$$

When the highest version of x is zero, we have $a_i = q_i$ as required.

3. Non-Recursive Base Conversion

We now present a theorem that derives non-recursive base conversion for a positive integer and a positive number less than one, respectively, with the result equivalent to a deterministic Markov process. The theorem proves that the Markov process is *twice* as efficient as standard base conversion.

Theorem I.

If $x = \sum_{k=1} b^{k-1} a_k$, for $0 \leq a_k \leq b-1$, $a_k \in \{0\} \cup \mathbb{Z}^+$ and $b \in \mathbb{Z}^+$, then

$$a_j = \left\lfloor \frac{x}{b^{j-1}} \right\rfloor - b \left\lfloor \frac{x}{b^j} \right\rfloor,$$

where $\lfloor \cdot \rfloor \equiv$ the greatest integer function (*i.e.* floor function) and $j, k \in \mathbb{Z}^+$.

Proof. First, evaluate $\frac{x}{b^j}$

$$x = \sum_{k=1} b^{k-1} a_k \quad (N \text{ operations})$$

$$\frac{x}{b^j} = \sum_{k \leq j} b^{k-1-j} a_k + \sum_{k > j} b^{k-1-j} a_k$$

Since the floor function is distributive,

$$\left\lfloor \frac{x}{b^j} \right\rfloor = \left\lfloor \sum_{k \leq j} b^{k-1-j} a_k \right\rfloor + \left\lfloor \sum_{k > j} b^{k-1-j} a_k \right\rfloor$$

Because of the condition: $0 \leq a_k \leq b-1$, only the second term within the first sum survives:

$$\left\lfloor \frac{x}{b^j} \right\rfloor = \sum_{k > j} b^{k-1-j} a_k$$

$$b \left\lfloor \frac{x}{b^j} \right\rfloor = \sum_{k > j} b^{k-j} a_k$$

$$\left\lfloor \frac{x}{b^{j-1}} \right\rfloor = \left\lfloor \sum_{k \leq j} b^{k-j} a_k \right\rfloor + \left\lfloor \sum_{k > j} b^{k-j} a_k \right\rfloor$$

Only the a_j term in the first sum survives {because of the geometric series} while the complete sum becomes:

$$\left\lfloor \frac{x}{b^{j-1}} \right\rfloor = a_j + \sum_{k > j} b^{k-j} a_k$$

$$b \left\lfloor \frac{x}{b^j} \right\rfloor = \sum_{k>j} b^{k-j} a_k$$

$$\left\lfloor \frac{x}{b^{j-1}} \right\rfloor - b \left\lfloor \frac{x}{b^j} \right\rfloor = a_j + \sum_{k>j} b^{k-j} a_k - \sum_{k>j} b^{k-j} a_k$$

$$\left\lfloor \frac{x}{b^{j-1}} \right\rfloor - b \left\lfloor \frac{x}{b^j} \right\rfloor = a_j \quad \square$$

Corollary:

Observe that this result is easily invertible:

$$\left\lfloor \frac{x}{a^{j-1}} \right\rfloor - a \left\lfloor \frac{x}{a^j} \right\rfloor = b_j$$

Example (1): First, convert 407 to base 4 using the standard recursion approach:

$$407/4 = 101 \rightarrow \text{remainder} = 3$$

$$101/4 = 25 \rightarrow \text{remainder} = 1$$

$$25/4 = 6 \rightarrow \text{remainder} = 1$$

$$6/4 = 1 \rightarrow \text{remainder} = 2$$

$$1/4 = 0 \rightarrow \text{remainder} = 1$$

Therefore, $407 = 12,113_{\text{four}}$ and this conversion required 10 (*i.e.* N) operations, which includes the remainder operations.

Now, use the non-recursive Markov approach:

$$\left\lfloor 407 \right\rfloor - 4 \left\lfloor \frac{407}{4} \right\rfloor = 3$$

$$\left\lfloor \frac{407}{4} \right\rfloor - 4 \left\lfloor \frac{407}{16} \right\rfloor = 1$$

$$\left\lfloor \frac{407}{16} \right\rfloor - 4 \left\lfloor \frac{407}{64} \right\rfloor = 1$$

$$\left\lfloor \frac{407}{64} \right\rfloor - 4 \left\lfloor \frac{407}{256} \right\rfloor = 2$$

$$\left\lfloor \frac{407}{256} \right\rfloor - 4 \left\lfloor \frac{407}{1024} \right\rfloor = 1$$

$$\left\lfloor \frac{407}{1024} \right\rfloor - 4 \left\lfloor \frac{407}{2048} \right\rfloor = 0$$

Therefore, $407 = 12,113_{\text{four}}$ and this conversion required 6 [*i.e.* $(N/2 + 1)$ operations.]

Example (2):

Convert 0.390625 to base 2 using the standard recursion approach:

$$0.390625 \times 2 = 0.78125 \rightarrow \text{remainder} \rightarrow 0$$

$$0.78125 \times 2 = 1.5625 \rightarrow \text{remainder} \rightarrow 1$$

$$0.5625 \times 2 = 1.125 \rightarrow \text{remainder} \rightarrow 1$$

$$0.125 \times 2 = 0.25 \rightarrow \text{remainder} \rightarrow 0$$

$$0.25 \times 2 = 0.5 \rightarrow \text{remainder} \rightarrow 0$$

$$0.5 \times 2 = 1 \rightarrow \text{remainder} \rightarrow 1$$

Therefore,

$$0.390625 = 0.011001_{\text{two}}$$

Now, use the non-recursive Markov approach, with $x = 0.390625$, $b = 2$ and $N = 6$.

$$a_1 = \lfloor 2^6 (0.390625) \rfloor - 2 \lfloor 2^5 (0.390625) \rfloor = 1$$

$$a_2 = \lfloor 2^5 (0.390625) \rfloor - 2 \lfloor 2^4 (0.390625) \rfloor = 0$$

$$a_3 = \lfloor 2^4 (0.390625) \rfloor - 2 \lfloor 2^3 (0.390625) \rfloor = 0$$

$$a_4 = \lfloor 2^3 (0.390625) \rfloor - 2 \lfloor 2^2 (0.390625) \rfloor = 1$$

$$a_5 = \lfloor 2^2 (0.390625) \rfloor - 2 \lfloor 2^1 (0.390625) \rfloor = 1$$

$$a_6 = \lfloor 2^1 (0.390625) \rfloor - 2 \lfloor 2^0 (0.390625) \rfloor = 0$$

Thus, we get the same result:

$$0.390625 = 0.011001_{\text{two}}$$

4. Conclusion

Coupling non-recursive base conversion to a deterministic Markov process involves defining states that represent each stage of the conversion, establishing deterministic transition rules that depend solely on the current state, and ensuring the process terminates in a final state representing the converted number. This approach leverages the deterministic nature of both the base conversion algorithm and the Markov process framework to achieve the desired coupling. With the presentation of Theorem I, in this paper, we prove that any recursive base conversion process can be converted into a Markov process with unit transition probabilities and the Markov process is twice as efficient as the standard recursive approach to base conversion.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

- [1] Matula, D.W. (1968) The Base Conversion Theorem. *Proceedings of the American Mathematical Society*, **19**, 716-723.
- [2] Hardy, G.H. and Wright, E.M. (1954) *An Introduction to the Theory of Numbers*. 3rd Edition, Clarendon Press, Oxford, England.
- [3] Robertson, J.E. (1958) A New Class of Digital Division Methods. *IRE Transactions on Electronic Computers*, **EC-7**, 218-222.

- <https://doi.org/10.1109/TEC.1958.5222579>
- [4] Wilson, J.B. and Ledley, R.S. (1961) An Algorithm for Rapid Binary Division. *IRE Transactions on Electronic Computers*, **12**, 662-670.
 - [5] Matula, D.W. (1967) Base Conversion Mappings. 1967 *Spring Joint Computer Conference, AFIPS Proceedings*, **30**, 311-318.
<https://doi.org/10.1145/1465482.1465532>
 - [6] Matula, D.W. (1967) Base Conversion Mappings. *American Federation of Information Processing Societies*, **30**, 311-318.
 - [7] Agrawal, D.P. (1975) Arithmetic Algorithms in a Negative Base. *IEEE Transactions on Computers*, **10**, 998-1000.
 - [8] DeRegt, M.P. (1967) Negative Radix Arithmetic. *Computer Design*, **6**, 52-63.
 - [9] Abramowitz, M. and Stegun, I.A. (Eds.) (1972) *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. 9th Printing, Dover, New York, p. 10.
 - [10] Arfken, G. (1985) *Mathematical Methods for Physicists*. 3rd Edition, Academic Press, Orlando, FL, 278-279.