The Collatz Conjecture: Structured Mathematical and Computational Study

Maurya Allimuthu
catchmaurya@gmail.com

June 2, 2025

1 Problem Statement

The **Collatz Conjecture** (also known as the 3n + 1 problem) posits:
For any positive integer n, repeated application of the rule

$$T(n) = \begin{cases} \frac{n}{2}, & \text{if } n \equiv 0 \pmod 2 \\ 3n+1, & \text{if } n \equiv 1 \pmod 2 \end{cases}$$

will eventually reach 1.

Despite its simple definition, no general proof or counterexample exists.

2 Analysis and Observations

2.1 Symbolic Equation Tracking

For small integers, symbolic manipulation reveals the sequence:

- $n = 8$:

$$8 \to 4 \to 2 \to 1 \Rightarrow 8 \cdot \left(\frac{1}{2}\right)^3 = 1$$

- $n = 3$:

$$3 \to 10 \to 5 \to 16 \to 8 \to \cdots \to 1$$

  Symbolically:

$$(3 \cdot ((3n+1)/2) + 1) \cdot \left(\frac{1}{2}\right)^4 = 1 \Rightarrow n = 3$$

2.2 Backward Construction Strategy

We define $T^{-1}(n)$ as:

- If n is even, then $T^{-1}(n) = 2n$
- If $n \equiv 4 \pmod 6$, then $T^{-1}(n) = \frac{n-1}{3}$

This leads to a reverse tree structure rooted at 1, where branches grow from known Collatz paths backward.

3 Approaches to the Problem

3.1 Dynamic Programming / Memoization

We compute and store the full Collatz paths for all $n \leq 1000$. If any number $x > 1000$ hits a known value during its path, we terminate the computation early.

Let $S = \{1, 2, \ldots, 1000\}$. If $T_i(x) \in S$, then $T(x) \to 1$ by inheritance.

3.2 Recursive Block Reduction

If the conjecture holds for $[1, 1000]$, then:

- $[1001, 2000]$: even numbers halve into ≤ 1000
- $[1001, 3001]$: many odd values produce ≤ 1000 via $(3n+1)/2$

Thus, the original range $[1, 1000]$ acts as a proof base, extending recursively over wider domains.

4 Proof Sketch

- Symbolic expressions for number paths
- Reverse tree modeling using $T^{-1}(n)$
- Cached values for efficient verification
- Block-wise reduction pattern

5 Graph / Tree Visualization

A tree structure rooted at 1 can be constructed:

- Nodes represent values n
- Edges represent Collatz transitions: $n \to T(n)$
- Backward expansion uses $T^{-1}(n)$ rules

Graph implementation can be performed using `networkx` and `matplotlib`.

6 Code Implementation

Python code computes paths and step counts for $n \in [1, 1000]$ with memoization. Example snippet:

```
collatz_cache = {}

def collatz_path(n):
    if n in collatz_cache:
        return collatz_cache[n]
    path = []
    original_n = n
    while n != 1:
        path.append(n)
        if n % 2 == 0:
            n = n // 2
        else:
            n = 3 * n + 1
```

```
path.append(1)
collatz_cache[original_n] = path
return path
```

7 Symbolic Equation Generator (Planned)

A tool can be developed to:

- Record each transformation symbolically
- Build nested expressions of the form:

$$n \to \frac{3n+1}{2} \to \cdots \to 1$$

8 Conclusion

We combine:

- Symbolic tracking of number reductions
- Reverse tree modeling ($T^{-1}(n)$)
- Dynamic programming for optimization
- Block-based recurrence strategy

This multi-layered approach blends theory and computation to progress toward a structured proof or reduction strategy for the Collatz Conjecture.