Joshua W Dunaitis 10/12/2025

Research Philosophy

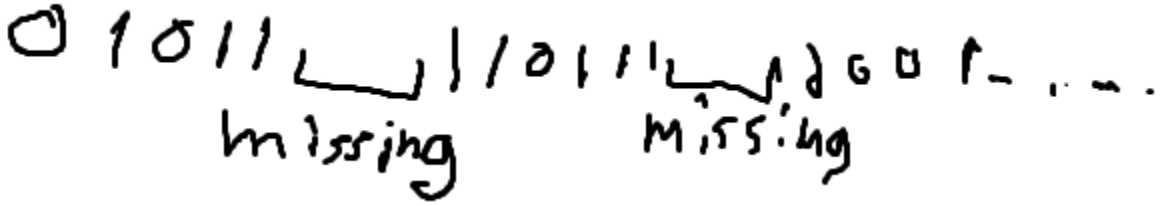
For my research, I start with a holistic model informed by reductionist findings. My modeling comes from my mathematics logic education and my economics training. I use a troubleshooting methodology informed by information technology and then an algorithmic analysis informed by computer programming. The algorithm is similar to a binary search. I consider this methodology “filling in the gaps in the sequence”.

I start with the beginning of existence on one side of the model and then have the current moment on the other side of the model.

I know many facts and theories (missing pieces). I start filling in my holistic existence model with all of the pieces to try to solve whatever I am interested in. As I fill in the pieces, I have large placeholders for missing pieces. I try to fit new pieces of information into those gaps. If they seem to fit, I have a working theory. If they are demonstrated not to fit by evidence (they are only demonstrated not to fit pending additional information) then I move the piece out of the model in that location and look to where else it may fit. I seek a different solution to the missing piece.

This holistic informed by reductionist model enables many theoretical solutions to problems that can later be refined or discarded with additional evidence.

Another way to view this model is as a binary sequence. I know what problem I am interested in and want to solve. I know the start of the binary sequence that I am interested in. I know much about the current state of the sequence. I try to fill in the gaps along the way.



These gaps can also be thought of as computer functions where they take some inputs, process them and an output is produced. These gaps are biological and cognitive programs.

This methodology allows for rapid iteration towards a solution.

References
Binary Search

Search result from Google

AI Overview

The search algorithm that repeatedly divides a list in half to find a value is called Binary Search.

How Binary Search Works:

- **Sorted List Requirement:** Binary search is only applicable to lists that are already sorted in ascending or descending order.
- **Find the Middle:** The algorithm starts by examining the element in the middle of the list.
- **Comparison:**
  - If the middle element is equal to the target value, the search is successful, and the position is returned.
  - If the middle element is less than the target value (for an ascending list), the algorithm knows the target value must be in the upper half of the list. The lower half is discarded.
  - If the middle element is greater than the target value (for an ascending list), the algorithm knows the target value must be in the lower half of the list. The upper half is discarded.
- **Repeat:** Steps 2 and 3 are repeated on the remaining half of the list until the target value is found or the search space becomes empty (meaning the value is not in the list).