I instinctively believe that everything that is happening in the world can be precisely interpreted by formulas and computable probability and vice versa. I guess this is the reason why I got immediately inspired when I was first introduced to the concept of Turing machines in ‘Introduction of the Theory of Computation’.

Problems can be turned into strings as Turing Machines and become computable. Including every detail in instructions, the state diagrams I drew in accordance with transition functions, however complex, always lead to the right answer. The elegance of computational theories with the rigorous logic and mathematical basis behind them fascinate me unrelentingly. Once as I turned variables into strings to determine whether my family will go to a specific restaurant for fun, a new question occurred to me: can I always write down these 7-tuples? In other words, can Turing machines solve every problem with hypothetically limitless steps?

Following the idea that the outputs are consistent for the same input, I began to consider probability. A-Level physics taught me that to get a random factor, the physical procedure needs to be included. Accordingly, the string variable that is independent of a physical factor can only generate pseudo-random number instead of a real random number, unless the start state is random.

On the other hand, there exists a logical solution to this, which grasped more of my attention. Assume that a Turing machine to decide acceptance between two strings of input can be constructed, and a second machine to give the opposite result can be built. Consequently, when Turing Machines themselves become the inputs, diagonalisation method rules out the possibility of the existence of the second machine by showing the existence of a paradox at the intersection, where the answer should be the opposite of itself. In no way can we even decide whether a certain input would be accepted.

The subtle self-referential construction in the Halting Problem’s second solution reminded me of the time I was a child, intrigued by the concept of “brain in the vat”, liar paradox and the equivalence between “if p then q” and “not p or q”. I can still remember how I nearly laughed the first time I found native set theory proved to be contradictory by Russell’s paradox, discovering that the intuitive principle “any definable collection is a set” cannot be true. Benefitting from continuously learning mathematical logic as years passed by, I gained further insight into abstract concepts. The similarity in construction when proving Gödel’s first incompleteness theorem shows me the charm of self-reference once more. Notwithstanding this, while I was researching the background of incompleteness theorem, concepts of axiomatic reasoning and formal systems illuminated the beauty of the areas of mathematics not covered by the school curriculum for me.

Compared to paradoxes, self-referential recursion functions differently. The concept of dynamic programming attracted me in the first place. The bottom-up approach is used to save the solutions of optimal subproblems in a table so that each overlapping problem only needs to be solved once. Top-down analysis is my favourite part, applying logical thinking to distinct situations and eventually solving the problems. Principles of algorithms gave me a new perspective when dealing with different tasks. Therefore, I have a desire to delve further into algorithms and to be prepared to think mathematically.

I find myself interested in almost every subject, from natural sciences to humanities. It took time to confirm what I crave for, but with the benefit of getting exposed to other captivating areas. It is always a pleasure to learn from people who conduct research in different areas.

I aspire to contemplate the logic behind the world and science by rational inference and the application of knowledge. The underlying principle of the world guides my curiosity and passion seamlessly.