Introduction of Constructive Mathematics

- In the last 30 years, with the growing influence of computer science, people want to extract an algorithm function to represent proof of existence.
- In addition, the issues of the standard decimal representation of the computation led to the idea of Constructive Mathematics. For example, when x = 0.3333...(infinite decimal places) multiply with 3. There is no way for the computer to decide the first digit is 0 or 1 because the computer can never decide the last digit. Moreover, the standard decimal representation cannot compare the real number. For the same x, the computer cannot precisely determine $x < \frac{1}{3}$ or $x \geq \frac{1}{3}$, because the computer cannot express x in fractional form and also cannot predict the next digit.
- Due to these theoretical disadvantages, Constructive Mathematics can be a good approach to describe real number values.
- In Constructive Mathematics, people should strictly construct phrases instead of interpreting “there exists.”

Basic Concepts in Constructive Mathematics

- **Definition 1: Alphabet** is a finite list of primitive symbols (letters). We use A to represent Alphabet.
- **Definition 2: An algorithm** can be thought of as a finite sequence of symbols from Alphabet.
- **Definition 3:** Let α be an algorithm in the alphabet A_1, and let α′ be an algorithm in the alphabet A_2. If α′ in A_2 with the exact same scheme as α. The algorithm α' will be called the **extension** of α.
- **Definition 4: A constructive Sequence of Natural Numbers(SNN)** is an algorithm transforming every natural number into a natural number. (N→N)
- **Definition 5: A Constructive Sequence of Rational Numbers(SRN)** is an algorithm transforming every natural number into a rational number. (N→Q)

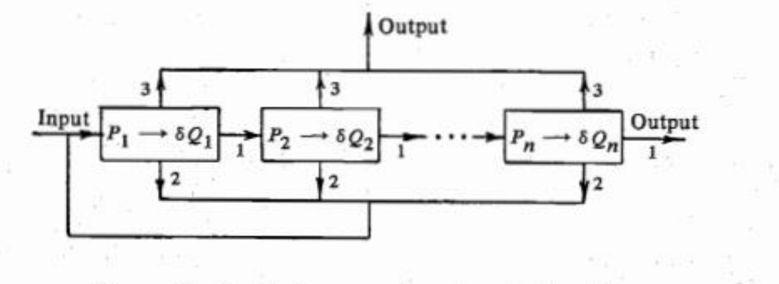

Figure 1: Block Diagram of a normal algorithm

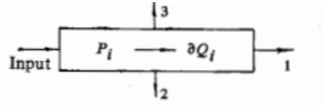

Figure 2: Each Step of a normal algorithm

(The figures are from the book, "Lectures on Constructive Mathematical Analysis")

Constructive Real Number

- **Definition 6: Constructive Real Number (CRN)** is a pair of program α and program β. An integer n is taken as the input. The program α gives the Cauchy convergent sequence $\alpha(n)$. $\beta(n)$is the constructive sequence of natural numbers such that for $\forall\ i, j \geq \beta(n)$,we have $|\ \alpha(i) - \alpha(j)| \leq 2^{-n}$. Under this situation, program α is fundamental, and program β(rate of convergence) is the regulator of the fundamentality of program α.
- Thus, π $(3.1415926\ldots)$ *and* e $(2.7182846\ldots)$ can be described in Constructive Mathematics.

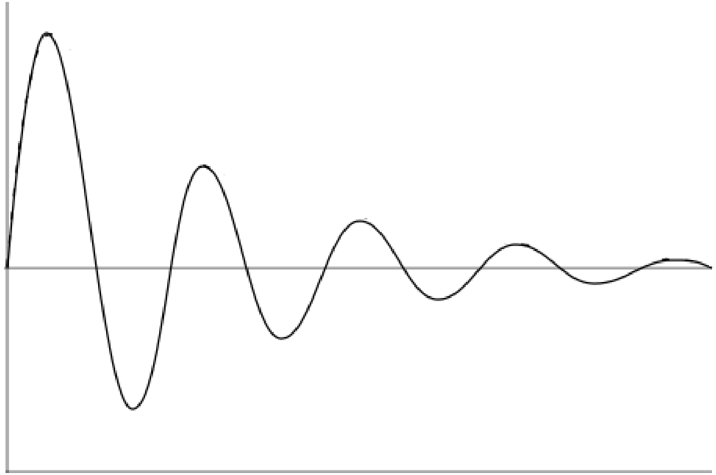

Figure 1: Constructive Real number

Properties of Computable Function

- **Definition 7**: A function is **computable** if there exists an algorithm that can do the same thing as the function.
- **Definition 8:** For $n, x \in N$, we defined that $U(n, x)$ is **a universal function** for a class of computable functions of one variable, if for every n, the function $U_n(x)$ is a computable function of one variable, and all computable functions of one variable are one of the U_n. $U_n(x)$ is defined as $U(n, x)$.
- **Lemma 1:** There exists a computable function U of two variables that is universal for the class of computable function of one variable.
- **Lemma 2:** There does not exist everywhere defined function of two variables that is universal for the class of computable everywhere defined function of one variable.
- **Lemma 3:** There exists a computable function that does not admit on everywhere defined computable extension.

Motivation

- In reality, people can’t obtain the exact number in some situations.
- For example, when we have a box that is 1m × 1m × 1m, measuring it up to a centimeter may get 0.99m × 1.01m × 1.00m. If we measure the box in millimeters, the result might be 0.991m × 1.009m × 1.001m. Therefore, when we get more precise measurements, the results will keep changing.
- People cannot obtain and describe the exact result by using traditional mathematics in this situation, but Constructive Mathematics can be an effective method to express it.
- Specifically, there might be some situations that the investments can not tell the exact answer, but we can get the profit closer and closer by doing a sequence of successive measurements. Thus, this idea helps me to generate my hypothesis.
- **Hypothesis: There is no computer program that is capable of always give the optimal way for investment from a few given investment options when the profits are constructive real numbers.**

Basic Settings and Theorem 1

- The entire project stems from the notion of a computer algorithm. The program we use in this project is a partially defined algorithm (Not all the natural numbers input can lead to an output) that is non-extendable to all the natural numbers. The program can have outputs either 0, 1, or never terminate.
- Lemma 1, Lemma 2, and Lemma 3 can prove the existence of partially defined, non-extendable algorithm, which is critical in proving the theorem.
- **Theorem 1: In Constructive Mathematics, there does not exists an algorithm that can always gives the optimal way of investment with the largest profits from two given investment options**.

Basic Settings for the Proof of Theorem 1

- We assume that there are two investable companies, called X and Y.
- The profit I can obtain from company X is the constructive real number a_n .
- The profit I can obtain from company Y is the constructive real number b_n .
- Let H be the non-extendable, partially-defined program that can generate a_n and b_n.
- We can define a_n and b_n according to the definition of Constructive Real Numbers and algorithm H.

Basic Settings of the Proof of Theory 1

- $a_{n,k} = 1$, if the computer program H has not finished on input n by the k^{th} step or if the program H has already finished working on n by the k^{th} step produced 1.
- $a_{n,k} = 1 + 2^{-m}$, if the computer program H has already finished on input n by the k^{th} step and produced 0. Variable m is the step number when it finished working on n.

- $b_{n,k} = 1$, if the computer program H has not finished working on input n by the k^{th} step or if the program H has already finished working on n by the k^{th} step produced 0.
- $b_{n,k} = 1 + 2^{-m}$, if the computer program H has finished on input n by the k^{th} step and produced 1. Variable m is the step number when it finished working on n.

Proof of Theory 1

- Here are the situations that the output of Algorithm H is 0 or 1:

- Output of Algorithm H is 0, $\begin{cases} a_n = 1 \\ b_n = 1 + 2^{-m} \end{cases}$ ⇒ $a_n > b_n$

- Output of Algorithm H is 1, $\begin{cases} a_n = 1 + 2^{-m} \\ b_n = 1 \end{cases}$ ⇒ $a_n < b_n$

- The profit of company X is larger than the profit of company Y when the output of Algorithm H is 0.
- The profit of company Y is larger than the profit of company X when the output of Algorithm H is 1.
- Note that, when algorithm H does not give an output, we can't compare a_n and b_n to determine the larger profit.

Proof of Theory 1

- We use the contradiction to prove theorem 1 by introducing a new hypothetical program P. Suppose that there exist an hypothetical program P that can always determine the optimal way for investment. Now we can prove that hypothetical program P will lead to the extension of algorithm H.
- Suppose that algorithm H will never terminate at x. According to our definition, algorithm P can obtain the optimal way of investment when the input is x, as it can always compare the constructive real numbers a_n and b_n.
- However, in this case, there would be an extension of algorithm H at x, which we called H', defined as follows:
    - H' = 1 when program P gives that $b_n > a_n$, company Y is profitable.
    - H' = 0 when program P gives that $a_n > b_n$, company X is profitable.
- Note that this is an extension of algorithm H at x, which can be all natural number inputs that is initially not defined in algorithm H, so H can be extended to all inputs. This extension lead to the contradiction with our non-extendible algorithm H.
- Thus, there would not be such computer program P that can always give the optimal investment solution from two options.

Generalization (Proof of Theorem 2)

- Now, we have already proved a non-existence algorithm that can choose the optimal solution from **two options**. However, we still want to prove that there is no algorithm that can obtain the optimal solution from **several investment options**.
- **Theorem 2: There is no algorithm that can obtain the optimal solution from several investment options.**
- Let us consider the condition that we have n investment options. Therefore, there are n constructive real numbers.
- When we compare two of the constructive real numbers, as we have already proved, there is no such algorithm that can choose the optimal investment option from these two choices. Otherwise, it will be extended to all-natural number inputs, which will lead to a contradiction.
- As a result, any of the two constructive real numbers cannot be compared. Eventually, the algorithm cannot give the investment option with the largest profit.
- Thus, we cannot find an algorithm to choose the optimal solution for a few given options of investment.

Main Results and Conclusion

- Overall, the general idea of this project is that if the program exist, program would be extendable, but the algorithm should be non-extendable. This lead to the contradiction to prove my hypothesis.
- We conclude the non-existence of the algorithm that can obtain the optimal solution for a few given options of investment.
- In the future, I would like to proceed in the following directions.
    - We have shown the non-existence of optimal solution for a few given options of constructive real number profits, but our proof doesn’t show whether we can acquire optimal solution for both real number profits and constructive real number profits.
    - In these few decades, scholars didn’t discover any effective approach to predict the stock market in Computer Science. Therefore, I think that Constructive Mathematics may be an tool to prove the non-existence of an algorithm to predict the stock market.