The Halting Problem: Why It's Unsolvable

111110542 林源茂

The **Halting Problem**, as proven by Alan Turing, states that there is no algorithm that can universally determine whether a given program will halt or run indefinitely for all possible inputs.

Proof Outline:

1. Assume there exists a program HaltChecker(program, input) that returns:
    - True if the program halts on the given input.
    - False if it runs indefinitely.
2. Construct a new program Paradox(program):

```
def Paradox(program):
    if HaltChecker(program, program):
        while True:
            pass
    else:
        return
```

3. Pass Paradox itself as input: Paradox(Paradox):
    - If HaltChecker(Paradox, Paradox) says it halts, Paradox enters an infinite loop.
    - If HaltChecker(Paradox, Paradox) says it doesn't halt, Paradox halts.
    - Contradiction! Hence, HaltChecker cannot exist.

A Heuristic Approach: Practical Attempt

111110542 林源茂

While solving the Halting Problem is impossible, we can create a program that uses **heuristics** to identify simple cases where a program will halt or run forever.