

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
```

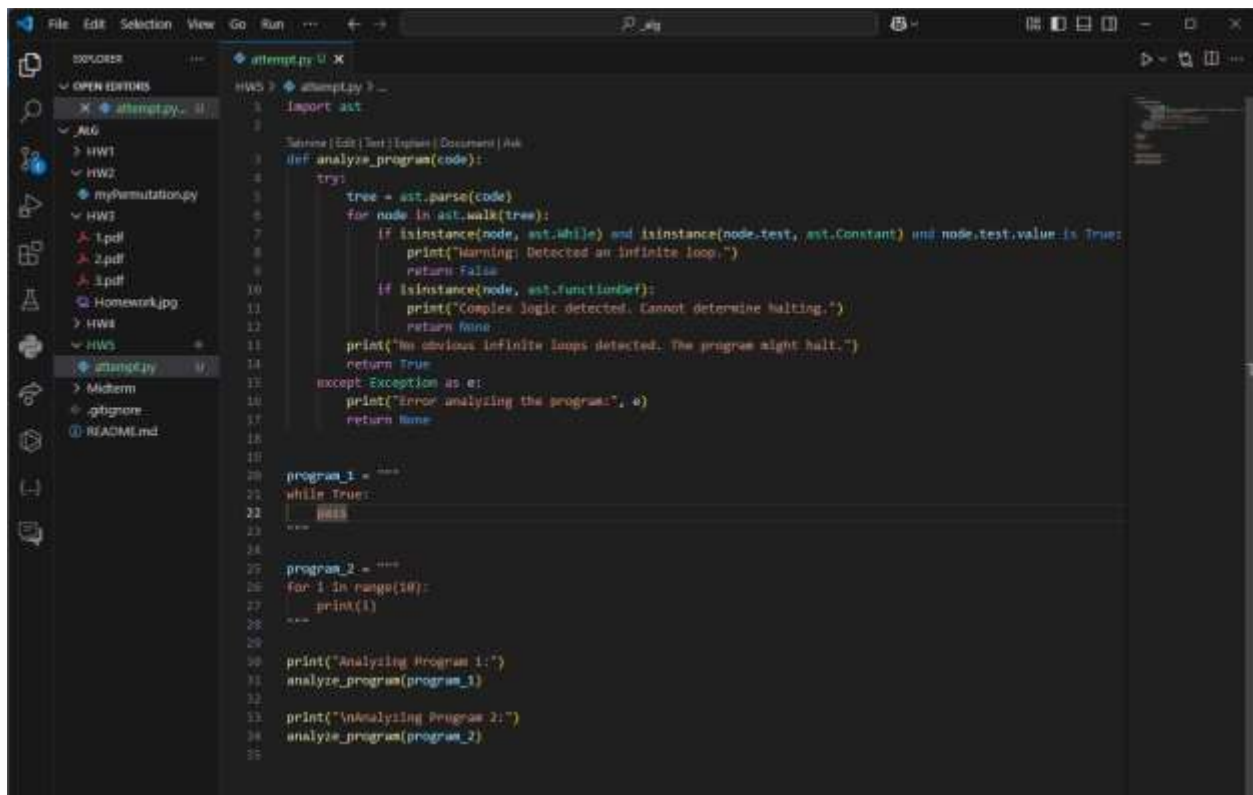
3. Pass `Paradox` itself as input: `Paradox(Paradox)`:

- 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.



```
File Edit Selection View Go Run ...  
HWS > attempt.py  
1 import ast  
2  
3 def analyze_program(code):  
4     try:  
5         tree = ast.parse(code)  
6         for node in ast.walk(tree):  
7             if isinstance(node, ast.While) and isinstance(node.test, ast.Constant) and node.test.value is True:  
8                 print("Warning: Detected an infinite loop.")  
9                 return False  
10            if isinstance(node, ast.FunctionDef):  
11                print("Complex logic detected. Cannot determine halting.")  
12                return None  
13            print("No obvious infinite loops detected. The program might halt.")  
14        return True  
15    except Exception as e:  
16        print("Error analyzing the program:", e)  
17        return None  
18  
19  
20 program_1 = """  
21 while True:  
22     pass  
23 """  
24  
25 program_2 = """  
26 for i in range(10):  
27     print(i)  
28 """  
29  
30 print("Analyzing Program 1:")  
31 analyze_program(program_1)  
32  
33 print("Analyzing Program 2:")  
34 analyze_program(program_2)  
35
```