Halting Problem

Will a program P
halt given input x?

```python
def foo(bar: List[int]) -> int:
    return len(bar)
```

```python
def baz(x):
    if x == 0:
        while True:
            pass
    else:
        return None
```

foo([1, 2, 3]) ✓

foo(1) ✗
↑ is not a List[int]

```java
int foo(List<Integer> bar){
    return bar.size();
}
```

TestHalt(P, x)
- "yes" if P(x) terminates in finite # of steps
- "no" if P(x) inf loops

TestHalt(baz, 1) ⟶ "yes"

TestHalt is uncomputable!

Proof: Assume TestHalt is computable.

```python
def Turing(P):
    if TestHalt(P, P) == "yes":
        while True:
            pass  # inf looping
    else:
        return None  # halt
```

Turing(Turing) halts. ✗

Turing(Turing) ∞ loops. ✗

TestHalt(Turing, Turing)

$(\text{TestHalt computable} \Rightarrow \text{Turing computable}) \wedge (\neg \text{Turing computable})$

$\Downarrow$

$\neg$ TestHalt computable ▨

Foo is not comp.

Pf: Use Foo to write TestHalt.

$(\text{Foo comp} \Rightarrow \text{TestHalt comp}) \wedge \neg \text{TestHalt comp}$

$\Downarrow$

$\neg$ Foo comp

7

COMPUTABILITY

Computer Science Mentors 70

Mar 08 - Mar 12
Spring 2021

Amogh Gupta, Sylvia Jin, Aekus Bhathal, Abinav Routhu, Debayan Bandyopadhyay

Roast us here: `https://tinyurl.com/csm70-feedback20`

Computability

1. Say that we have a program M that decides whether any input program halts as long as it prints out the string "ABC" as the first operation that it carries out. Can such a program exist? Prove your answer.

```
def foo(x):
    print("ABC")
    return 3
```

M(foo, 100) = False

M is not comp.

```
def TestHalt(P, x):
    def P'(x):
        print ("ABC")
        P(x)
    if M(P', x) == True:
        return "halts"
    else:
        return "loops"
```

2. (a) Is it possible to write a program that takes a natural number n as input, and finds the shortest arithmetic formula which computes n? For the purpose of this question, a formula is a sequence consisting of some valid combination of (decimal) digits, standard binary operators ($+$, $\times$, the "^" operator that raises to a power), and parentheses. We define the length of a formula as the number of characters in the formula. Specifically, each operator, decimal digit, or parentheses counts as one character.

(*Hint:* Think about whether it's possible to enumerate the set of possible arithmetic formulas. How would you know when to stop?)

$n = 98763351$

$n = 100000$

$0-9, +, \times, \wedge$

100000

$10 \wedge 5$

10×10000

(b) Now say you wish to write a program that, given a natural number input n, finds another program (e.g. in Java or C) which prints out n. The discovered program should have the minimum execution-time-plus-length of all the programs that print n. Execution time is measured by the number of CPU instructions executed, while "length" is the number of characters in the source code. Can this be done?

(*Hint:* Is it possible to tell whether a program halts on a given input within t steps? What can you say about the execution-time-plus-length of the program if you know that it does not halt within t steps?)

n 10000

print("10000")

14

$14 + 46 = \underbrace{60} = \ell$

$60 - \ell$

3. (a) Explain why the notion of the "smallest positive integer that cannot be defined in under 280 characters" is paradoxical.

(b) Prove that for any length n, there is at least one string of bits that cannot be compressed to less than n bits.

(c) Say you have a program K that outputs the Kolmogorov complexity of any input string. Under the assumption that you can use such a program K as a subroutine, design another program P that takes an integer n as input, and outputs the length-n binary string with the highest Kolmogorov complexity. If there is more than one string with the highest complexity, output the one that comes first lexicographically.

(d) Let's say you compile the program P you just wrote and get an m bit executable, for some $m \in \mathbb{N}$ (i.e. the program P can be represented in m bits). Prove that the program P (and consequently the program K) cannot exist.

(*Hint*: Consider what happens when P is given a very large input n.)