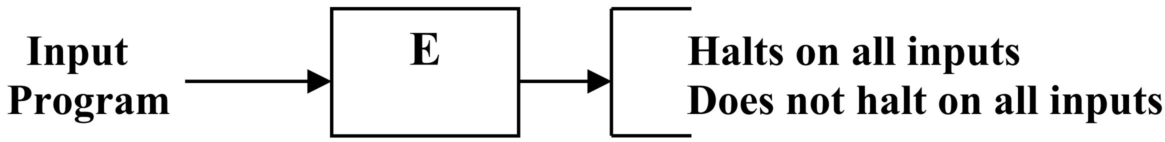
NONCOMPUTABLE PROBLEMS

Example 1: Halting Problem



<u>Program E</u>: able to read any program and halt after a finite time with the correct answer: either the given input program halts on all inputs or it does not.

```
procedure halt(P, x, result)
   begin
    Body of the halt routine.
    if ... then result := `Halts.`
    else result := `Does not halt.`
   end


procedure selfhalt(P, result)
   begin
    halt(p, p, answer)
    if answer = `Halt.` then
        result := `Halts on self.`
    else result := `Does not halt on self.`
   end


procedure contrary(p)
   begin
    selfhalt(p, answer)
    if answer = `Halts on self.` then
        while true do
            anwer := `x`
   end
```

Example 2: Functions

Assume: we have the natural numbers in the table.

X	$F_1(X)$	$F_2(X)$	$F_3(X)$	...	$F_i(X)$	...	$F_e(X)$
1	1	1	2				1 + 1 = 2
2	2	4	4				4 + 1 = 5
3	3	9	6				6 + 1 = 7
...	...	...					
i	$F_1(i)$	$F_2(i)$	$F_3(i)$		$F_i(i)$		$F_i(i) + 1$
...							

Assume: the table contains every function that can be considered on these numbers.

Question: F_e is a member of the table or not.
Answer: NOT.

The set of functions that have positive integers for inputs and outputs is uncountable.

There exist functions, which cannot be computed algorithmically.

INTRACTABLE COMPUTATION

A computation is <u>intractable</u> if its execution time increases with increasing *n* faster than any polynomial.

Example 1: Towers of Hanoi

n	t(approximate)
5	0.17 sec
10	5.62 sec
15	3.00 min
30	68.23 days
35	5.98 years
55	6267840 years
70	205385000000 years

$$t = 5.49 * 10^{-3} * 2^n$$

Example 2: Traveling Salesman Problem

Find the shortest route:

- **Find all routes.**
- **Compute their length.**
- **Select the optimum one.**

Number of cities = n, $n > 2$

$$S = \frac{1}{2}(n - 1)!$$

S: number of different paths.