For Loops are not enough - we want to prove:

Thm: There exists a computable fnc f which is not P.R.

Search for something → is while loop.

2 Proofs:
1) with more detail, but $f(n)$ will be useless
2) with less detail, but $f(n)$ will be useful

we need some auxiliary notion for the proof: the code of a p.r. fnc.

(i) A p.r. fnc can be described as an expression $PR(\sigma, \pi_1^3 \circ \sigma)$

↓ latex

(ii) PR(\sigma, \pi^3-1| \circ \sigma)

↓ ASCII

(iii) 1011 01 . . . 1 . . . .

↓ append 1 in front bcz we're dealing with non-neg int

101101 . . . → interpret it as an int, this is called the code of a p.r. fnc

Lemma:

There exists an algorithm that, given a natural number c,

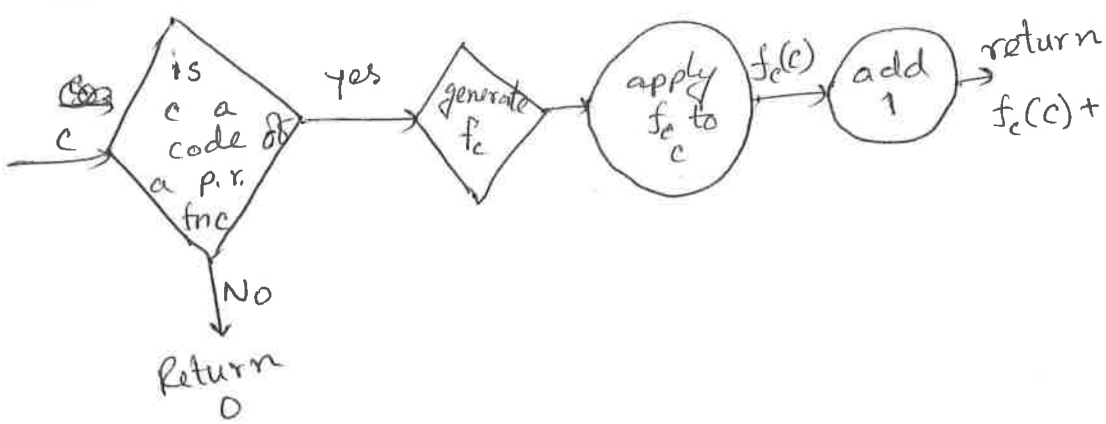
* checks whether c is a code of a p.r. fnc
* if yes, returns the executable file computing this fnc.

this file will be denoted f_c.

Let us construct a function $f(c)$ about which we will prove that it is computable but not p.r.

$$f(c) = \begin{cases} f_c(c) + 1 & \text{if } c \text{ is a code of a p.r. fnc} \\ 0 & \text{otherwise} \end{cases}$$

Let's prove that $f(n)$ is computable.



To prove the theorem, we now need to show that $f(c)$ is not p.r.

we will prove it by contradiction. we assume that $f(c)$ is p.r., and we will get a contradiction.

Let c_0 be a code of p.r. fnc $f(c)$.

$\cancel{\forall n (f_{c_0}(n))}$

$\forall c \; (f_{c_0}(c) = f(c))$

Q: what is the value of $f_{c_0}(c_0)$?

This is true for all n, in particular for $c = c_0$

$$f_{c_0}(c_0) = f(c_0).$$

On the other hand, by def^n of $f(c)$,

$$f(c_0) = f_{c_0}(c_0) + 1$$

$$\cancel{f_{c_0}(c_0)} = \cancel{f_{c_0}(c_0)} + 1$$

$0 = 1$, which is a contradiction.

This proves that $F(c)$ is not p.r.

$$f(c) = \begin{cases} f_c(c) + 1 & \text{if } c \text{ is a code of a p.r fnc} \\ 0 & \text{otherwise.} \end{cases}$$

Informal explanation based on a toy example:

$f_0(n) = 0$

1 is not a code

$f_2(n) = n + 1$

$f_3(n) = n^2$

$f_4(n) = 2 * n$

Let's form a table

$f_n \backslash n$	0	1	2	3	4	...
f_0	$\boxed{0}$	0	0	0	0	
f_1	—	$\boxed{—}$	—	—	—	—
f_2	1	2	$\boxed{3}$	4	5	...
f_3	0	1	4	$\boxed{9}$	16	...
f_4	0	2	4	6	$\boxed{8}$	...
⋮						
f_n	1	0	4	10	9	...

We are using only the diagonal element, so it's called diagonal construction.

George Cantor proves that, set of real numbers cannot be countable.

$n+1$

$$a+b = a + \underbrace{1 + \cdots + 1}_{b \text{ times}}$$

$$a * b = \underbrace{a + a + \cdots + a}_{b \text{ times}}$$

$$a^b = \underbrace{a * \cdots * a}_{b \text{ times}}$$

$${}^{b}a = a^{a^{a^{\cdots a}}} \Big\} b \text{ times}$$

Archimedes

0-th order $f_0(a,b) = a+1$

1-st order $f_1(a,b) = a+b$

2-nd order $f_2(a,b) = a * b$

3-rd order $f_3(a,b) = a^b$

4-th " $f_4(a,b) = f_3\underbrace{(a, f_3(a, \ldots))}_{b \text{ times}} = a^{a^{a^{a^{\cdot^{\cdot}}}}}$

$$f_4(a,0) = 1$$
$$f_4(a,m+1) = f_3(a, f_4(a,m))$$
$$f_{k+1}(a,m+1) = f_k(a, f_{k+1}(a,m))$$

Each of these operations is p.r.

Ackermann describes the fnc
$$A(n) = f_n(n,n)$$

<u>Thm:</u> $A(n)$ is not p.r.

<u>Hint:</u>
$$A(0) = f_0(0,0) = 0+1 = 1$$
$$A(1) = f_1(1,1) = 1+1 = 2$$
$$A(2) = f_2(2,2) = 2*2 = 4$$
$$A(3) = f_3(3,3) = 3^3 = 27$$
$$A(4) = f_4(4,4) = \cancel{f_3(4, \ldots)}\ 4^{4^{4^4}} = 4^{10^{153.6}}$$

$$2^{512} = (2^{10})^{51.2} = (10^3)^{51.2}$$