17.2.09.

For halt checker proof: [what is f_n ? & what is code?]

* Java program — is a sequence of ASCII symbols.
* In the computer, every ASCII symbol is a seq of 0's & 1's.

! ___ ! ___ ← put 1 in front.

get a natural #, this # is called a code of the Java program.

what is f_c ?

Let's start with a natural # c.

we strip off 1 in front.

f_c → Java program corr. to # c.

$$f(n) = \begin{cases} f_n(n)+1 & \text{if } n \text{ is a valid code \& } f_n \text{ halts on } n. \\ 0 & \text{otherwise.} \end{cases}$$

[Not all computable function is A-PR.

To prove this we will construct a function $f(n)$.

ⓐ which is computable.

ⓑ but not A-PR.

ⓐ To prove the 1st part, our goal is to assign a n...

Let's start with an expression like. $PR(\pi_1^1, A(n))$.

with Latex. PR(Pi ^1-1, A(n)) .]

Turing m/c :-

Projection: $\pi_2^2 = (n_1, n_2) = n_2$.

we need to move the 1's like this :-

we want: #|1|1|#|1|1|1|#

start. return

→ |1|#|#|#|1|1|1|#

finally #|1|1|1|#

(start, #) → (R, 1, erasing 1)

(erasing 1, 1) → (erasing 1, #, R)

(erasing 1, #) → (fwd 2, R)

(fwd 2, 1) → (fwd 2, R)

(fwd 2, #) → (backtrack, L)

(backtrack, 1) → (carry 1, #, L)

(carry 1, #) → (checknextblank, L)

(carry 1, 1) → (carry 1, L)

|1|#|#|#|1|1|#

carry 1

(check next blank, 1) → (final cleaning, #, R)

(final cleaning, #) → (final move, 1, L)

(final move, #) → halt

[order of these rules doesn't matter]

[h/w : π_2^3]

{ (check next blank, #) → (go Right, R)
  (go Right, #) → (1, R, fwd 2)