If true, # | a | # | # | 1 | #
we return it by π_2^3 which is 0 ($m=0$)

if false, we increase m value by 1.

then we get the state

| a | # | 1 |

Now we copy again, $\underbrace{\# \mid a \mid \# \mid 1 \mid \#}$ $\underbrace{a \mid \# \mid 1 \mid \#}_{\text{apply } P(a,1)}$

true: # | a | # | 1 | # | 1 | #
return 1.

false: # | a | # | 1 | 1 | #
Increase m by 1. → then copy & repeat the steps.

Finally we will get μm for which $P(a,m)$ is true.

next thursday – test.

Thursday 26th Feb

Big picture: μ-recursive fn (church); TM (Turing)

⇓

Proving that every μ-rec fn is Turing-computable.

$$\mu m\, P(\vec{a}, m) = f(\vec{a}) \qquad \vec{a} = (a_1, \dots a_k)$$

we know that, $P(\vec{a}, m)$ is turing computable. we want: produce a TM for computing $f(\vec{a})$

| $\vec{a}$ | # | m

we start with

Analyze the problem :–

idea: try $m=0$; if $P(\vec{a},0)$ return 0, else try $m=1$

if $P(\vec{a},1)$ return 1.

∴ $m=0$;

while $(!P(\vec{a},m))$

{ m++; }

If we start with # | $\vec{a}$ | # | # if $P(\vec{a},0)$ false

then # | # | # so a is lost.

Thus, copy the state $(\vec{a},0)$ once.

correct way:

| a | # |

step 1: copy # | $\vec{a}$ | # | # | $\vec{a}$ | # | #

| # | $\vec{a}$ | # | # | $\vec{a}$ | # | # |

P

$P(\vec{a},0)$ true

$P(\vec{a},0)$ false

| # | $\vec{a}$ | # | # | 1 | # |

we return 0 by π^{k+2}_{k+1} i.e. π^3_2 here.

| # | $\vec{a}$ | # | # | # |

going right & see # means false.

increase m by 1.

| # | $\vec{a}$ | # | 1 | # |

copy again.

2. possibilities; $P(\vec{a},1)$ true, $P(\vec{a},1)$ false

| # | $\vec{a}$ | # | 1 | # | $\vec{a}$ | # | 1 | # |

P

| # | $\vec{a}$ | # | 1 | # | 1 |

return m = 1 by π^{k+2}_{k+1}

| # | $\vec{a}$ | # | 1 | # | # |

increase by 1.

| # | $\vec{a}$ | # | 1 | 1 | # |

then again copy

| # | $\vec{a}$ | # | 1 | 1 | # | $\vec{a}$ | # | 1 | 1 | # |

P

& repeat steps again

Thus finally we get, m for which $P(\vec{a},m)$ is true.

$\therefore$ le-recursion is Turing-computable.

Problem with what we had so far:-

+ : when we had a -ve result it was ok.

– : sometimes we actually produced algorithms, but many of these algorithms were unrealistically long.

Example: $\sigma(n) = n+1$

If n = 1 million, then no of operations on TM would be very huge.

For security, we use integers $\approx$ 100 - 200 digits long.

impossible # of steps.

we need to separate "theoretical" algorithms from "practical" algorithms.

feasible

How do we define an algo?

1930s — TM
— μ-recursion

How we define a feasible algorithm? No perfect definition exists.

Existing definion:

sorting $O(n \log n)$. Exhaustive search 26^n → not feasible.

linear search $O(n)$.

matrix multiplication $O(n^3)$

A little bit of physics: $\approx 6 \cdot 10^{17}$ secs.

$T \approx$ 20 billion years $\approx 2 \cdot 10^{10}$ years (timeline of universe).

we can make, $\dfrac{T}{\Delta t}$ → smallest possible time interval.

computational steps

1 year = 365 days $\approx 4 \cdot 10^2$ days $\approx 4 \cdot 10^2 \cdot 2 \cdot 10^1 \cdot 6 \cdot 10^1 \cdot 6 \cdot 10^1$

1 day = 24 hrs $\approx 2 \cdot 10^1$ hrs $\approx 300 \cdot 10^5 \approx 3 \cdot 10^7$ secs.

1 hr = 60 min $\approx 6 \cdot 10^1$ min

1 min $\approx$ 60 sec $= 6 \cdot 10^1$ sec.

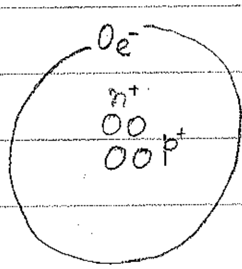
1 GHz $\approx 10^9$ computer then $6 \cdot 10^{23}$ operations

1 THz $\approx 10^{12}$ " " $6 \cdot 10^{26}$.

1 PHz $\approx 10^{15}$ " " $6 \cdot 10^{29}$

$\Delta t = \dfrac{\text{dist}}{\text{velo}} = \dfrac{}{\cancel{c \to \text{speed of light}}}$

time during which the light goes through the smallest elementary particle.



electron, nutron, proton have size because of Heisenberg's Uncertainty Principle.

↳ locatn & velocity of particle cannot be measured at same time.

$\Delta x \cdot \Delta p \geq \hbar$ ← planck's constant.

$\Delta p \to mv$.

So, $\Delta x \geq \frac{\hbar}{m v}$ → velocity of particle.
→ mass of particle

proton's size is smallest.

$$\Delta t = \frac{\text{Size of a proton}}{c \text{ (speed of light)}} \approx 10^{-23} \text{ secs.}$$

$\Delta x \approx 10^{-13}$ cm, $c \approx 3 \times 10^5$ km/sec.

Thus, computational steps that can be done during T → $\frac{6 \cdot 10^{17}}{10^{-23}} \approx 6 \cdot 10^{40}$

10^{80} normal particles
10^{90} particles overall

} If we consider all the particles are computing . then 10^{130} steps

So, $26^n = 10^{130}$. $\qquad (10^{1.5})^n \approx 10^{130}$.

$$n = \frac{130}{1.5} \approx 87.$$

[So, input size 87 for exhaustive search]