Halting Problem

On the basis of configuration of a Turing Machine, the output of TM can be

1) Halt : The machine starting at this configuration will halt after a finite number of states.

Accept and machine M reaches q_f, then the input string w is accepted by M.

Reject If M doesn't reach q_f, then the input string w is not accepted by machine M.

2) No halt : The machine starting at this configuration never reaches a halt state, no matter how long it runs.

Now the Question is:
Given any functional matrix, input data tape and initial configuration, then it is possible to determine whether the process will ever halt ?
This is called halting problem.

That means we are asking for a procedure which enable us to solve the halting problem for every pair (machine, tape)

No, It's non-decidable.

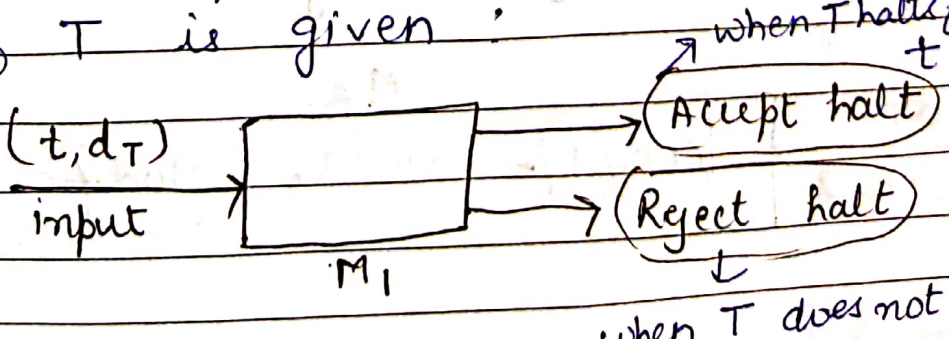
→ It means given a Turing machine will it halt, when given particular I/p string?

→ Given a program written in some language (C, C++, Python), will it ever get into infinite loop or will it always terminate?

Ans It's non-decidable. In general we can't always know ie Run and check.

Halting problem is unsolvable.

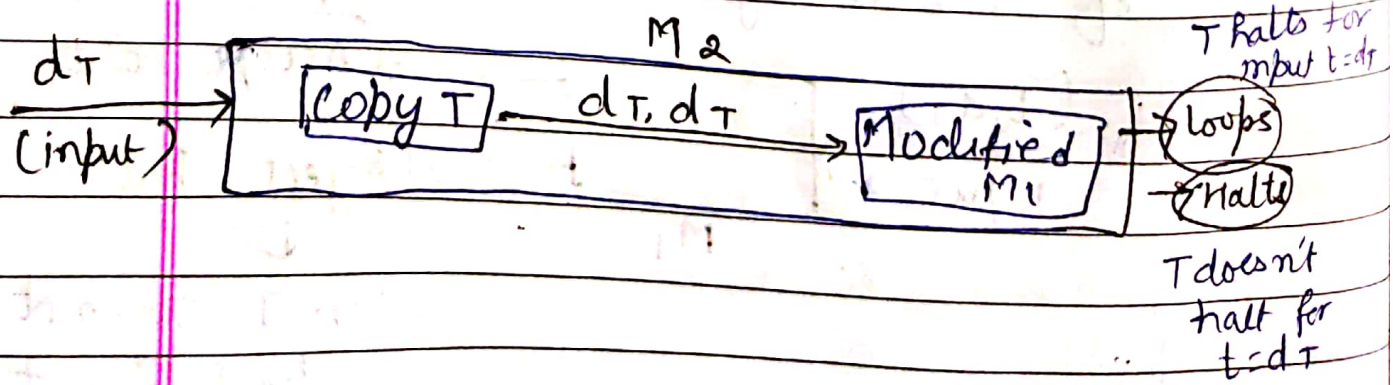
Let, there exists a TM M_1 which decides whether or not any computation by a TM T will ever halt, when a description (d_T) of T and tape t of T is given:



Accept halt: Then for every input (t, d_T) to M_1, if T halt for input t. M_1 also halts, which is called accept halt.

Reject halt: If T does not halt for input t, then the M_1 will halt which is called reject halt

Now, we will consider another TM M_2 which takes an input d_T. It first copies d_T and duplicates d_T on its tape and then this duplicate tape information is given as input to machine M_1. But machine M_1 is a modified machine with the modification that whenever M_1 is supposed to reach an accept state, M_1 loops forever. Hence behaviour of M_2 is as given. It loops if T halts for input $t = d_T$ and halts if T doesnot halt for $T = d_T$. The T is any turing machine.



Date
Page
KRISHNA

Given a program will it halt?
Can we design a machine, which it given a program can find out or decide if that program will always halt ?

Let's assume yes we can

H(P, i)
- Halt
- No halt

Let us write a program

```
C(Y)
{
  if (H(Y,Y) == Halt)
    loop forever;   {while { };}

  else
    return;
}
```

C(C)

H(Y,Y) == Halt → Not halt (loop)

H(Y,Y) = not ha → Halt.

This contradict our assum