<u>High-level</u> description of a TM:

"On input ___ [optional description]:
1. ___
2. ___
3. ___
   a) ___
   b) ___
4. ___ "

If no description, the input is an arbitrary string (over the input alphabet of the TM)

Otherwise, the input is assumed to be a syntactically correct string encoding the described object.

<u>Ex</u>: Graph reachability algo

"On input $\langle G, s, t \rangle$ where G is a digraph and s & t are vertices of G:
1. Using BFS from s, visit all vertices reachable from s
2. If (or when) t is found in step (1), then accept.
3. Reject"

Things you can do:

- simulate a TM M on some input w
- simulate a TM M on some input w for up to t steps (t: another input)

<u>Eg.</u>: Let N be the following TM:

N = "On input $\langle M, w, t \rangle$, where M is a TM, w a string, and t is a natural number:
1. Run M on input w for t steps (or until it halts, whichever comes first)
2. If M accepts w within t steps, then accept; else reject."

N is a decider, even if M isn't.

<u>Def</u>: Let

$A_{TM} := \{\langle M, w \rangle : M \text{ is a TM, } w \text{ a string, and } M \text{ accepts input } w\}$

"The acceptance problem for TMs"

<u>Thm</u>: A_{TM} is undecidable.

<u>Proof</u>: Assume otherwise. Let D be a TM deciding A_{TM}.

Consider the following TM

F := "On input $\langle M \rangle$, where M is a TM:
1. Construct the string $\langle M, \langle M \rangle \rangle$
2. Run D on input $\langle M, \langle M \rangle \rangle$
3. If D accepts, then reject; else accept."

Consider F running on input $\langle F \rangle$:
1. Form the string $\langle F, \langle F \rangle \rangle$
2. Run D on input $\langle F, \langle F \rangle \rangle$
3. If D accepts then reject; else accept.

D accepts $\langle F, \langle F \rangle \rangle \Rightarrow$ F accepts $\langle F \rangle$ (assumption about D)

$\therefore$ F rejects $\langle F \rangle$ (by def of F) $\lightning$

Similarly D rejects $\langle F, \langle F \rangle \rangle$ then F accepts $\langle F \rangle$.

In either case