Undecidability Reductions

$A_{TM} = \{\langle M, w\rangle \mid M \text{ is TM and } M \text{ accepts } w\}$

- Fact: A_{TM} is undecidable.

$HALT_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM \& } M \text{ halts on input } w\}$.

Theorem 5.1: $HALT_{TM}$ is undecidable

Proof: Suppose not. $HALT_{TM}$ has a decider R. (R is an oracle (genie)). we will use R to show that A_{TM} is also decidable, leading to a contradiction.

"R" On input $\langle M, w\rangle$.
If M accepts w or rejects w, accept.
If M loops on w, reject.

Here is a decider S for A_{TM}.

S = "On input $\langle M, w\rangle$:
- Run R on $\langle M, w\rangle$ to see if M halts or loops.
- If R rejects, it means M loops, so reject.
- If R accepts, it means M halts. So, it is safe to simulate M on w.
  If M accepts w, accept.
  If M rejects w, reject."

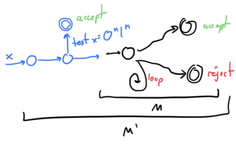
S is a decider for A_{TM}, a contradiction. □

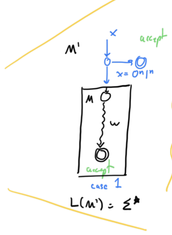
$REG_{TM} = \{\langle M\rangle \mid M \text{ is a TM \& } L(M) \text{ is regular}\}$.

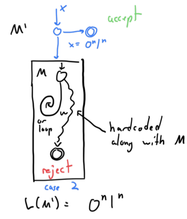
Theorem 5.3: REG_{TM} is undecidable.

Proof: Suppose not. It has a decider R.
We construct a TM to decide A_{TM}.

S = "On input $\langle M, w\rangle$:
1. Modify M to create M'.







Don't run M'! Just input M' into R to tell us if L(M') is regular or not.
- If L(M') is regular, then M accepts w, accept.
- If L(M') is not regular, then M rejects w. reject"

S is a decider for A_{TM} a contradiction. □

$Output = \{\langle P, x, s\rangle \mid$ Program\TM P on input x will output s at any time.$\}$

"Outputs" can mean stdout, print to tape, etc.

Theorem: Output is undecidable

Proof: Suppose not. Let R be decider (oracle) for Output.
Let $\langle M, w\rangle$ be an instance of HALT.

S = " On any input x, s:
Simulate M with input w.
If the simulation halts:
output s.
Else:
Loop forever & never output s."

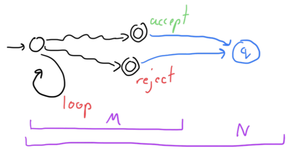
Run R on input $\langle S, x, s\rangle$.
If S output s, then M halted on w.
Otherwise, then M did not halt on w.
We have decided $\langle M, w\rangle$, so we can decide any instance of HALT, a a contradiction. □

$REACH = \{\langle M, w, q\rangle \mid$ TM M enters vertex q while running input w$\}$.

Theorem: REACH is undecidable

Proof: Suppose not. Let O be a decider (oracle) for REACH.
Let $\langle M, w\rangle$ be an instance of HALT.

N = " On input x, q:
1. Simulate M on input w:
2. If M halts, transition to state q.
3. If M does not halt, never transition to q."



Input $\langle N, x, q\rangle$ into O.
If O decides that N entered state q, then M halted on w.
If not, then M did not halt on w.
We have decided $\langle M, w\rangle$, so we can decide HALT, a contradiction □.