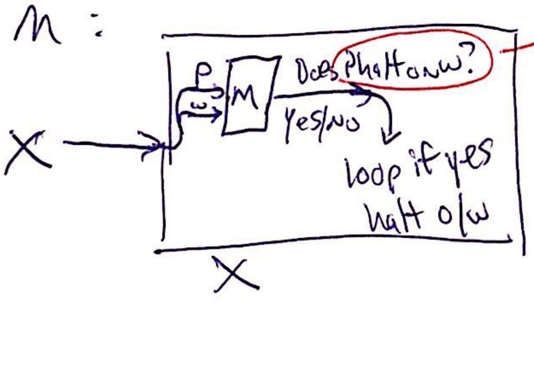
The halting problem: Given a program P and an input string w, will P ever halt if run on input w

So $H = \{ \text{"}(P,w)\text{"} \mid P \text{ eventually halts when invoked on input } w \}$

"_" here means a reasonable string representation for...

Thm. No program correctly $\underbrace{\text{decides membership in } H}_{\text{decides } H}$.

Pf. Suppose program M decides H. Build contradictory program X out of M:



Does X halt on X?

Lemma: X halts on "X" iff M says X does not halt on X

Contradicts M deciding H.

Consider $H_\lambda = \{ \text{"}P\text{"} \mid$ Program P halts eventually when provided empty string λ as input $\}$

to show H_λ undecidable (i.e. no program decides H_λ), we can use a <u>reduction</u> from H.

Show how to use a tester for H_λ to solve problems in H

<u>Formally</u> H reduces to H_λ if I can provide a computable converter τ on strings ($\tau: \underbrace{\Sigma^*}_{\text{strings}} \to \underbrace{\Sigma^*}_{\text{strings}}$) that convert H questions to H_λ questions:

$$\forall x \in \Sigma^* \quad x \in H \text{ iff } \tau(x) \in H_\lambda$$

$$(P, w) \in H \text{ iff } P' \in H_\lambda$$

τ converts (P, w) to P' where P' runs P on w but takes no input

Thm if undecidable language L reduces to Language L' then L' is also undecidable.