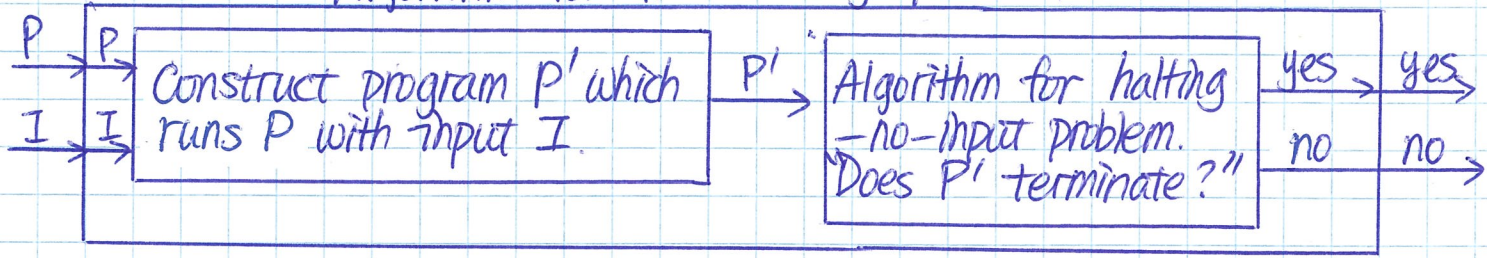Theorem: The halting-no-input problem is undecidable.

The halting-no-input problem: Given a program P that takes no input, does P terminate?
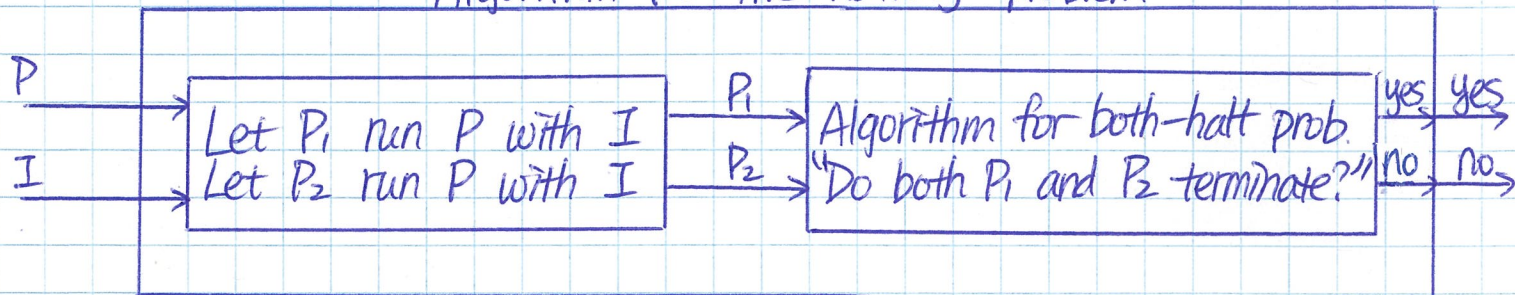
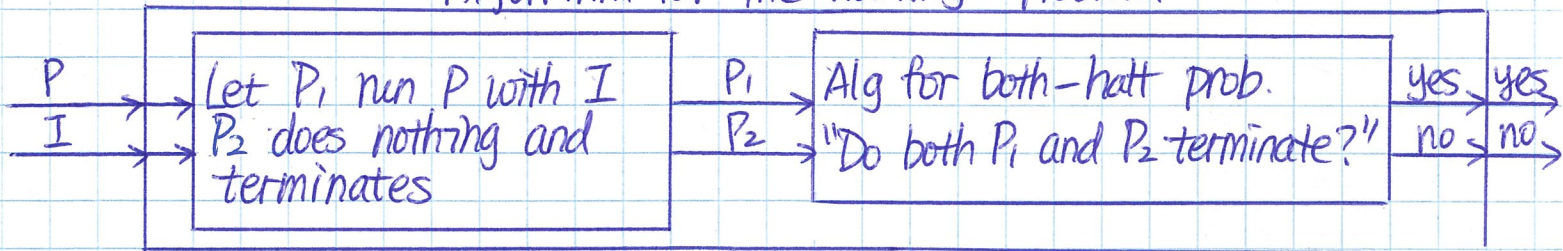"Does P terminate with input I?"
Algorithm for the halting problem

P → P → | Construct program P' which runs P with input I | → P' → | Algorithm for halting -no-input problem. "Does P' terminate?" | → yes → yes
I → I → | | | | → no → no

Theorem: The both-halt problem is undecidable.

The both-halt problem: Given two programs P_1 and P_2, do both programs terminate?

Reduction #1: "Does P terminate with input I?"
Algorithm for the halting problem

	Let P_1 run P with I Let P_2 run P with I	P_1 P_2	Algorithm for both-halt prob. "Do both P_1 and P_2 terminate?"	yes no	yes no
P I					

Reduction #2 *Does reduction #2 lead to a valid proof?*
"Does P terminate with input I?"
Algorithm for the halting problem

P I	Let P_1 run P with I P_2 does nothing and terminates	P_1 P_2	Alg for both-halt prob. "Do both P_1 and P_2 terminate?"	yes no	yes no

Reduction #3 *Does reduction #3 lead to a valid proof?*
"Does P terminate with input I?"
Algorithm for the halting problem

P I	Let P_1 run P with I Let P_2 run forever	P_1 P_2	Alg for the both-halt problem "Do both P_1 and P_2 terminate?"	yes no	yes no