18 April 2025	Program	Analysis
· · · · · · · · · · · · · · · · · · ·		· · · · · · · · · · · · · · · · · · ·
Plan		
* Program Equivale		
· · · · · · · · · · · · · · · · · · ·		
$EQ \notin RE$		
EQ ¢ CORE		
* Announcements		
* TIVIVIOUNCEMENIS		
* Rice's Theorem		

co-Recognizalde coRE		Recognizulde RE
	R	
E E E E E E E E E E E E E E E E E E E	Solvable in finite tin Jer D sti	$x \in L \implies D  \text{accepts}  \times \\ \times \notin L \implies D  \text{vejects}  \times \\$

co-Recognizalde coRE	Recognizable RE
Decidalde R	$= \begin{bmatrix} z \\ z$
Solvable in finite Educider D set	$fime$ $x \in L \implies D  accepts  \times$ $x \notin L \implies D  vejects  \times$

	Recognizulde
co-Recognizalde coRE	RE A A A A A A A A A A A A A A A A A A A
Decid	alle J program P
	$s \in f(P) = L$
Solvable in Edecideu D	finite fine $x \in L \implies D \ accepts \times X \notin L \implies D \ vejects \times X$

Recognizable RE co-Recognizalde CORE Decidable 94 

Recognizable RE co-Recognizalde CORE DIAG DIAG Decidable Z { P > : P does not accept { P > ] DIAG P accepts {P} ? DIAG

Recognizable RE co-Recognizalde CORE DIAG DIAG HAL Decidable Q halts on input x HALT =  $\left\{ \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \end{array} = \left\{ \begin{array}{c} 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \end{array} = \left\{ \begin{array}{c} 1 \\ 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \end{array} = \left\{ \begin{array}{c} 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \\ 1 \end{array} = \left\{ \begin{array}{c} 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \end{array} = \left\{ \begin{array}{c} 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \end{array} = \left\{ \begin{array}{c} 1 \end{array} = \left\{ \begin{array}{c} 1 \end{array} \right\} = \left\{ \begin{array}{c} 1 \end{array} = \left\{ \begin{array}{c} 1 \end{array} \\ 1 \end{array} = \left\{ \begin{array}{c} 1 \end{array} \\ 1 \end{array} = \left\{ \begin{array}{c} 1 \end{array} = \left\{ \end{array} = \left\{ \begin{array}{c} 1 \end{array} = \left\{ \begin{array}{c} 1 \end{array} = \left\{ \left\{ \begin{array}{c} 1 \end{array} = \left\{ \end{array} = \left\{ \left\{ \begin{array}{c} 1 \end{array} \right\} = \left\{ \left\{ \end{array} = \left\{ \begin{array}{c} 1 \end{array} = \left\{ \left\{ \end{array} \right\} = \left\{ \left\{ \begin{array}{c} 1 \end{array} = \left\{ \end{array} = \left\{ \left\{ \begin{array}{c} 1 \end{array} \right\} = \left\{ \left\{ \end{array} = \left\{ \end{array} = \left\{ \end{array} = \left\{ \left\{ \end{array} \right\} = \left\{ \end{array} = \left\{ \\ \end{array} = \left\{ \left\{ \end{array} = \left\{ \end{array} = \left\{ \end{array} =$ HALT = does not halt on input X

EQ. Recognizable co-Recognizalde CORE DIAG DIAG HAI Decidable  $\left\{ \left\langle P \right\rangle \# \left\langle Q \right\rangle \right\}$ :  $\mathcal{L}(P) = \mathcal{L}(Q)$ F Given two programs, do they recognize the same language?

· · · /	Ann			nert	· · · ·	· ·	· · · ·			· ·	• •	· ·	· ·	· · · ·		•	•	· ·	• •	• •	· · · ·	
· · · · ·		· · ·	· · · ·			Jue	 	tpri		20 		<ul> <li>.</li> <li>.&lt;</li></ul>	· · · · · · · · · · · · · · · · · · ·	Kîp.				    	    		· · · · · · · · · · · · · · · · · · ·	
· · · ·			· · · ·	Preliv	· · · · · · Z		vecy	rade	2 2 4	ج	رەرە	 २८९२ 	2. d.		n s N s	. ~	ر جـ د	Ker		· · ·	· · · ·	
· · ·	· · · ·	· ·	· · · ·	· ·	· · · ·	· ·	· · ·	· ·	• •	· ·	· ·	· ·	· ·	· · · ·	· ·	• •	•	· ·	· ·	· ·	· · ·	
· · ·	· · · ·	· ·	· · · ·	· ·	· · · ·	· ·	· · ·	· ·	• •	· ·	· · ·	· ·	· · ·	· · · ·	· ·	• · ·	•	· ·	· ·	· · ·	· · ·	
				· ·																		
· · ·	· · ·	· ·	· · · ·	· ·	· · · ·	· ·	· · ·	• •	• •	• •	• •	• •	· ·	· · ·	· ·	• •	•	• • • •	• •	• •	· · ·	
				· ·						• •	• •					• •	•	• •	• •	• •		

$EQ = \{\langle P \rangle \# \langle Q \rangle : \mathcal{L}(P) = \mathcal{L}(Q) \}$
Given two programs, do they recognize the same language?
Theorem. EQ & RE U coRE. La Not Recognizable, nor coRecognizable!
·       ·
1 1
· · · · · · · · · · · · · · · · · · ·

$EQ = \{\langle P \rangle \# \langle Q \rangle : \mathcal{L}(P) = \mathcal{L}(Q) \}$
Given two programs, do they recognize the same language?
Theorem. EQ & RE U coRE. La Not Recognizable, nou correcognizable!
Proof Approach: Reduction from the Halting Problem.

Recall.	HALTE	coRE	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · ·	Determin	ing if Q h	alts on input x
	is ve	cognizable, b	rt not decidable.
· · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · ·	· · · · · · · · · · ·	· · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · ·	· · · · · · · · · · · ·	· · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·
			· · · · · · · · · · · · · · · · · · ·
			· · · · · · · · · · · · · · · · · · ·

Recall. HALT & CORE
La Determining if Q halts on input x is recognizable, but not decidable.
TS VECOUT DUT DUT DE L'ANDEL
$HALT \leq EQ \implies EQ \notin GRE$ .
· · · · · · · · · · · · · · · · · · ·
Computable Reduction R
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·
· · · · · · · · · · · · · · · · · · ·

Recall. HALT &	z coRE	· · · · · · · · · · · · · · · · · · ·
Deterro is	vining if Q recognizable,	halts on input x but not decidable.
$HALT \leq E$		$Q \notin G RE$ .
Computable	Reduction R.	.       .
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	$\left\langle \begin{array}{cccccccccccccccccccccccccccccccccccc$
Q halts on x		$\mathcal{I}(\mathcal{P},) = \mathcal{I}(\mathcal{P},)$
Q does not halt on x		$L = (P_{0}) + f + L(P_{1})$

Reduction from HALT to EQ. Given (Q,x), write descriptions of P., P. as on input w Province and a second sec on input w  $Output \langle P, \rangle \neq \langle P, \rangle$ 

Reduction from HALT to EQ. Given {Q,x}, write descriptions of P., P. as on input w ignore W Accept ·Pr · · · on input w Output  $\langle P_n \rangle \# \langle P_n \rangle$ 

Reduction from HALT to EQ. Given (Q,x), write descriptions of P., P. as on input w ignore N Accept. / Hand code on input W ignore w, and run Q on x description of  $Q^{2} Q^{2} X \times Q^{2}$ Accept  $Output \langle P_{n} \rangle \# \langle P_{n} \rangle$ 

Reduction from HALT to EQ. Given (Q,x), write descriptions of P., P. as  $\mathcal{I}(P_{o}) = \sum^{*}$ on input w jgnove N Accept. on input w ignore w, and run Q on x  $(P, r) = \frac{1}{2} \cdot (P, r) = \frac{1}{2}$ Accept Output (P) # (P)

Pi on input w ignore w, and run Q on x	.       .
Accept	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
$\mathcal{L}(\mathcal{P}, \mathcal{I}) = \mathcal{I} \otimes \mathcal{I} \otimes \mathcal{I}$	
(Q,x) E HALT => Q halts on x in finite time	· · · · · · · · · · · · ·
$\langle Q, X \rangle \in HALT \implies Q$ halfs on $X$ in finite time	.       .
	.       .
	.         .

Pi on input w ignore w, and run Q on x Accept	<ul> <li>.</li> <li>.&lt;</li></ul>
$\mathcal{L}(\mathcal{P}_{1}) = \mathcal{J} \otimes \mathcal{W} \in \mathbb{Z}^{\times}$ . $\mathcal{P}_{1}$ accepts $\mathcal{W}$	· · · ·
$\langle Q, x \rangle \in HALT \implies Q$ halfs on x in finite time $\implies \forall w \in \Sigma^{\times}, P, accepts w$	· · · ·
1       1	
· · · · · · · · · · · · · · · · · · ·	

Pi on input w ignore w, and run Q on x	· · · · · · · · · · · · · · · · · · ·
$\mathcal{L}(\mathcal{P}, \mathcal{I}) = \mathcal{T} \otimes \mathcal{L}(\mathcal{P}, \mathcal{I})$	· · ·
$\langle Q, \chi \rangle \in HALT \implies Q$ halfs on $\chi$ in finite time	· · ·
$\implies \forall w \in \Sigma^{\times}, P_{i} \text{ accepts } W$ $\implies \qquad \qquad$	· · ·
	• •
	• •

Pi on input w ignore w, and run Q on x Accept Accept $X(P, ) = Z w \in Z^* : Pi accepts w J$
$\langle Q, X \rangle \in HALT \implies Q$ halfs on $X$ in finite time $\implies \forall w \in \Sigma^{X}, P_{i} \text{ accepts } W$ $\implies \mathcal{I}(P_{i}) = \Sigma^{X}$
$\langle Q, X \rangle \notin HALT \implies Q \text{ does NOT hulf on } X$ $\implies \forall w \in \mathbb{Z}^{*}, P, \text{ does not half on w}$ $\implies \mathcal{I}(P_{i}) = \emptyset$

$HALT \longrightarrow EQ$	· · · · · · · · · · · · · · · · · · ·
$Q^{n}$ halts on $X^{n}$	$\Leftrightarrow \qquad f(P_n) = f(P_n) = \sum^*$
·       ·	
.       .	
· · · · · · · · · · · · · · · · · · ·	
$\begin{array}{cccccccccccccccccccccccccccccccccccc$	HALT
$E \otimes E \otimes$	.       .
·       ·	

HALT EQ	· · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · ·	.       .       .       .       .       .       .         .       .       .       .       .       .       .       .         .       .       .       .       .       .       .       .         .       .       .       .       .       .       .       .         .       .       .       .       .       .       .       .	•
Q does not halt on X		$f(P_{n}) = f(P_{n})$		.     .     .     .     .     .       .     .     .     .     .     .       .     .     .     .     .       .     .     .     .     .	•
· · · · · · · · · · · · · · · · · · ·	· · · · · · · · ·	· · · · · · · · · · · · ·	· · · · · ·	.       .       .       .       .       .       .         .       .       .       .       .       .       .       .         .       .       .       .       .       .       .       .         .       .       .       .       .       .       .       .         .       .       .       .       .       .       .       .	•
· · · · · · · · · · · · · · · · · · ·	· · · · · · · ·	· · · · · · · · · · · ·	· · · · · ·	.       .       .       .       .       .       .         .       .       .       .       .       .       .       .         .       .       .       .       .       .       .       .         .       .       .       .       .       .       .       .	•
		RE	· · · · · ·	 	•
		HALT	· · · · · ·		•
			· · · · · ·	· · · · · · · ·	•
			· · · · · ·	· · · · · · · ·	•
$\Rightarrow$ EQ $\notin$ RE					•
					•

Recall, HALT & R		
$H_{A}$		
	e Reduction	.       .
$\left( \begin{array}{c} \begin{array}{c} \\ \\ \\ \end{array} \end{array} \right) = \left( \begin{array}{c} \\ \\ \\ \end{array} \right) = \left( \begin{array}{c} \\ \end{array} \right) = \left( \begin{array}{c} \\ \\ \end{array} \right) = \left( \begin{array}{c} \end{array} \right) =$		
Q does Not halt on x		$\mathcal{L} (\mathcal{P}, \mathcal{P}) = \mathcal{L} (\mathcal{P}, \mathcal{P})$
Q helts on x		
·       ·		

Reduction from HALT to EQ. Given (Q,X), write descriptions of P, P, as Por a a on input w · P. · · · · on input w Output  $\langle P_{o} \rangle \# \langle P_{i} \rangle$ 

Reduction from HALT to EQ. Criven (Q,X), write descriptions of Po, P, as  $\int_{-\infty}^{\infty} \left( P_{a}^{b} \right) = 0 \quad (a)$ on input w ignore w Reject  $\int (P_{1}) = ?$ ignove w, and run Q on x Accept)  $Output \langle P_{o} \rangle \# \langle P_{i} \rangle$ 

Prinder W ignore w, and run Q on X (Accept)
$\langle Q, X \rangle \in HAUT \implies Q \text{ does not halt on } X$ $\implies L(P, ) = \phi$
$\langle Q, X \rangle \in HAUT \implies Q$ halts on $X$ $\implies \mathcal{I}(P_1) = \Sigma^{X}$
$\mathcal{L}(\mathcal{P}_{0}) = \mathcal{L}(\mathcal{P}_{1}) = \phi$ ( $\Rightarrow$ ) Q does not halt

 $EQ = \frac{7}{\langle P_{o} \rangle} # \langle P_{i} \rangle : \mathcal{L}(P_{o}) = \mathcal{L}(P_{i})$ EQ & RE U CORE No way to prove or refite two programs have the same functionality

The Check-GPT Problem. * Write an inefficient algorithm A for 4820 homework * Ask GPT to return an efficient algorithm A* that solves the same problem as A. * Return True iff A and A* solve the same problem.
Theorem. Check - GPT is Undecidable!
Namely, there is no algorithm (current or future) that can reliably check the output of AI for correctness

	angu	<u>a</u> ge	s ak	pont	Pro	grang	<u>}                                    </u>			 	
 ★	- Ex	amp	les a								
				HAL		Q ,	~	· · · · · ·	  	· · · · · ·	· · · · · ·
a a A A A A	Ma		such	lan	guaze	s ar		decida	66.	· · · · · ·	· · · · · ·
· · ·	· · ·	· · ·	· · · ·	· · ·	· · · · · ·	· · · · · ·	· · · · ·	· · · · · ·	· · · · · ·	· · · · · ·	· · · · · ·
· · ·		·	have	+	Show			eduction	for	each	
• • •											
· · ·	· · ·	· · ·	· · · · ·	· · ·	· · · · · ·		· · · · · ·	· · · · · ·		· · · · · ·	· · · · · ·
· · · ·	· · · ·	· · · ·	· · · · ·	· · · ·	· · · · · · ·	· · · · · · ·	· · · · · ·	· · · · · ·	· · · · · ·	· · · · · · ·	  
· · · ·	· · · ·	· · · ·	<ul> <li>.</li> <li>.&lt;</li></ul>	· · · ·	· · · · · · ·	· · · · · · ·		<ul> <li>.</li> <li>.&lt;</li></ul>	· · · · · · ·	· · · · · · ·	  
<ul> <li>.</li> <li>.&lt;</li></ul>	<ul> <li>.</li> <li>.</li></ul>	<ul> <li>.</li> <li>.&lt;</li></ul>	.         .         .           .         .         .           .         .         .           .         .         .           .         .         .           .         .         .           .         .         .           .         .         .           .         .         .           .         .         .           .         .         .	<ul> <li>.</li> <li>.</li></ul>	.     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .	· · · · · · ·		.       .       .       .         .       .       .       .         .       .       .       .         .       .       .       .         .       .       .       .         .       .       .       .         .       .       .       .         .       .       .       .         .       .       .       .         .       .       .       .         .       .       .       .         .       .       .       .         .       .       .       .	· · · · · · ·	.     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .	.         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .
<ul> <li>.</li> <li>.&lt;</li></ul>	<ul> <li>.</li> <li>.&lt;</li></ul>	<ul> <li>.</li> <li>.&lt;</li></ul>	.         .         .           .         .         .           .         .         .           .         .         .           .         .         .           .         .         .           .         .         .           .         .         .           .         .         .           .         .         .           .         .         .	<ul> <li>.</li> <li>.</li></ul>	.         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .	· · · · · · ·		.         .         .         .           .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .	· · · · · · ·	.     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .       .     .     .     .     .	.         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .
								<ul> <li></li> <li></li></ul>	· · · · · · · ·	.         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .	.         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .
• • •		• • •							· · · · · · · ·	.         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .         .           .	.         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .         .
· · · ·	· · · ·	· · ·	· · · · ·	· · ·	· · · · · ·	· · · · · ·	· · · · · ·	· · · · · ·	.         .	.         .	.         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .
· · · ·	· · · ·	· · · ·	· · · · ·	· · · ·	· · · · · ·	· · · · · ·	· · · · · ·		.         .	.         .	.         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .
· · · ·	· · · · · · · ·	  	  	· · · · · · · ·	· · · · · · ·	· · · · · · ·		· · · · · ·	.         .	.         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .           .         .         .         .         .         .         .         .         .           .	.         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .           .         .         .         .         .         .

Semantic Languages \* A language of Program encodings is semantic if  $\langle P \rangle \in L$  is determined by the language  $\mathcal{J}(P)$ . input - output behavior  $\cdot \cdot \circ f \cdot \cdot P \cdot$ 

Semantic Languages \* A language of Program encodings a l is semantic if  $\langle P \rangle \in L$  is determined by the language  $\mathcal{J}(P)$ . Example Semantic Languages  $\mathcal{L}_{EMPTT} = \left\{ \left\langle P \right\rangle \right\} : \mathcal{L}(P) = \phi \left\{ \right\}$  $L_{ALL} = \left\{ \left\langle P \right\rangle : \mathcal{J}(P) = \mathcal{J}^{*} \right\}$ Linite = { { P} · [ Z(P)] is finite } Non-Semantic Language L four-loop = Z (P): Puses a four loop g detail

Rice's Theorem $\longrightarrow L \notin [\Phi, Z^*]$
Suppose L'is a nontrivial semantic language.
Then, L is undecidable.
· · · · · · · · · · · · · · · · · · ·
Implication for Program Analysis / Security.
Rice's Theorem $\implies$ Impossible to prove correctness / security
Rice's Theorem $\implies$ Correctness / security of arbitrary programs.
Rice's Theorem $\Longrightarrow$ correctness / security
Rice's Theorem $\Longrightarrow$ correctness / security
Rice's Theorem $\implies$ correctness / security of arbitrary programs.

Proof Reduction from HALT. is non-trivial => ] ] some (M) E L (similar argument) for case \$L Suppose ( on input w, Reject.) & [ Reduction Given (Q,x), construct P as · <del>P</del> · Output (P)

Proof Reduction from HALT. s non-trivial  $\implies$  F some  $\langle M \rangle \in L$ (similar argument) for case &L) Suppose ( on input w, Reject) & ! Reduction Given (Q,x), construct P as on input w. run Qon X run. M. on w. if Maccepts, Accept) Output (P)

Proof Reduction from HALT. is non-trivial >> ] some (M) EL (similar argument) for case &L) Suppose ( on input w, Rejecti) & Reduction Given (Q,x), construct P as  $\mathcal{I}(P) = \mathcal{I}(M) \Longrightarrow \langle P \rangle \in \mathbb{I}$ on input w. run Qonx (Q, X) E HALT run Mon w if M accepts, (Accept) Output (P)

Proof Reduction from HALT. ⇒ J some (M) E L L is non-trivial (similar argument) for case &L) Suppose ( on input w, Reject) & Reduction Given (Q,x), construct P as  $\mathcal{J}(P) = \mathcal{J}(M) \Longrightarrow \langle P \rangle \in$ on input w run Q on X {Q, x} E HALT run Mon w if Maccepts, (Accept)  $\mathcal{I}(P) = \phi \implies \langle P \rangle \neq 1$ Output (P) (Q,X) & HALT