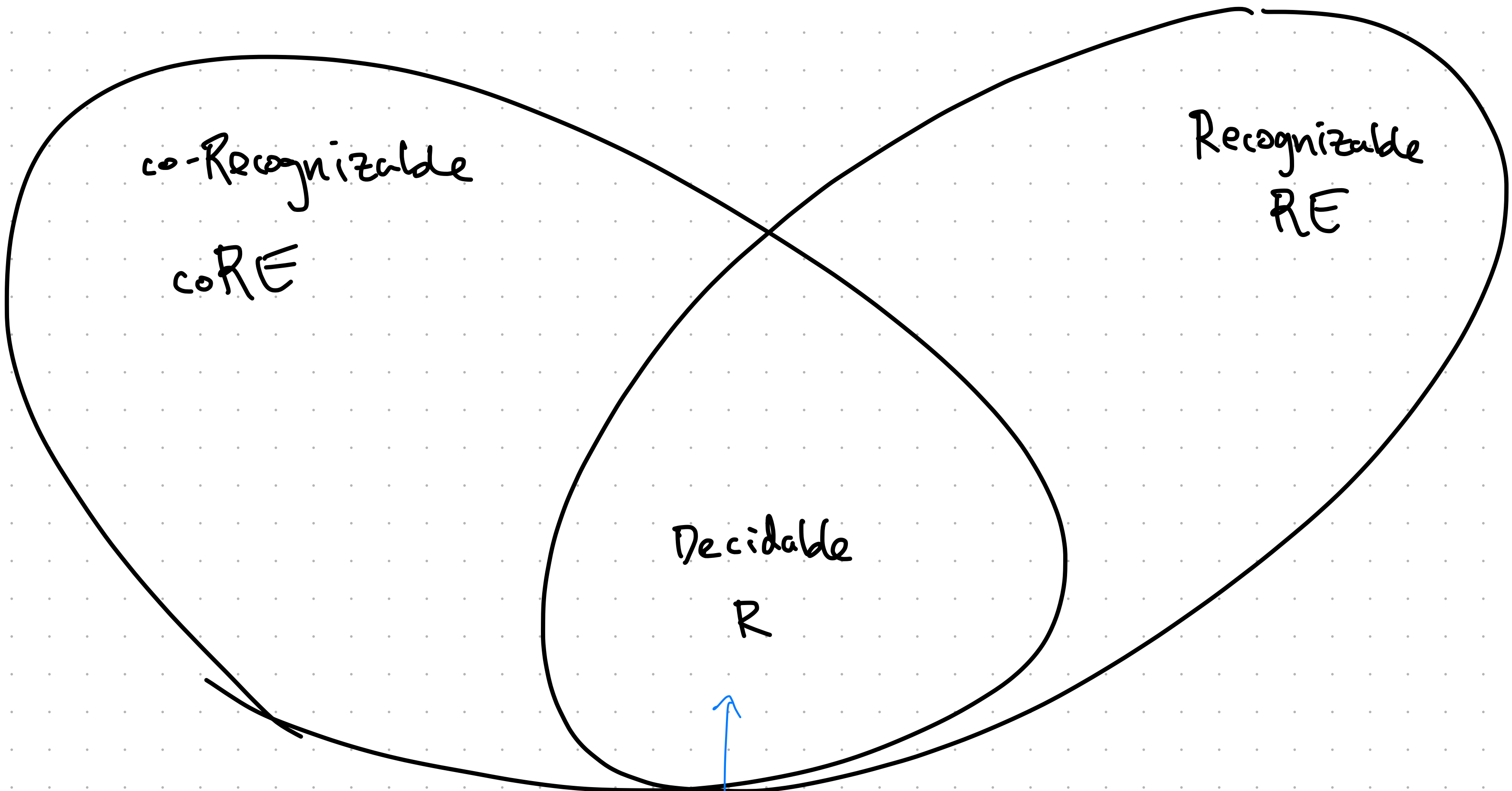
18 April 2025   Program Analysis

Plan

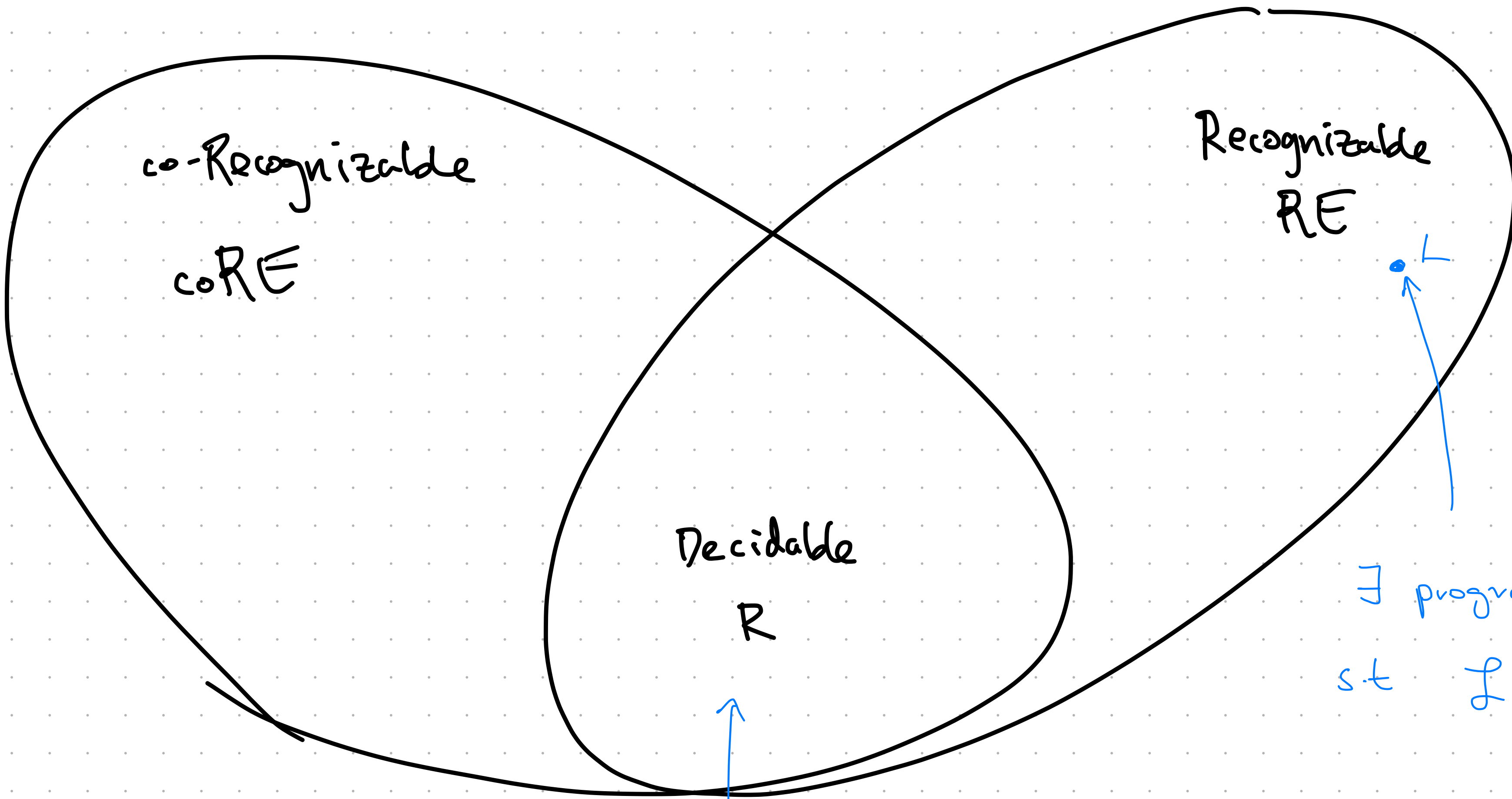
* Program Equivalence
  - ↳ $EQ \notin RE$
  - ↳ $EQ \notin coRE$
* Announcements
* Rice's Theorem



Solvable in finite time

$\exists$ decider D s.t. $x \in L \Rightarrow D$ accepts x

$x \notin L \Rightarrow D$ rejects x



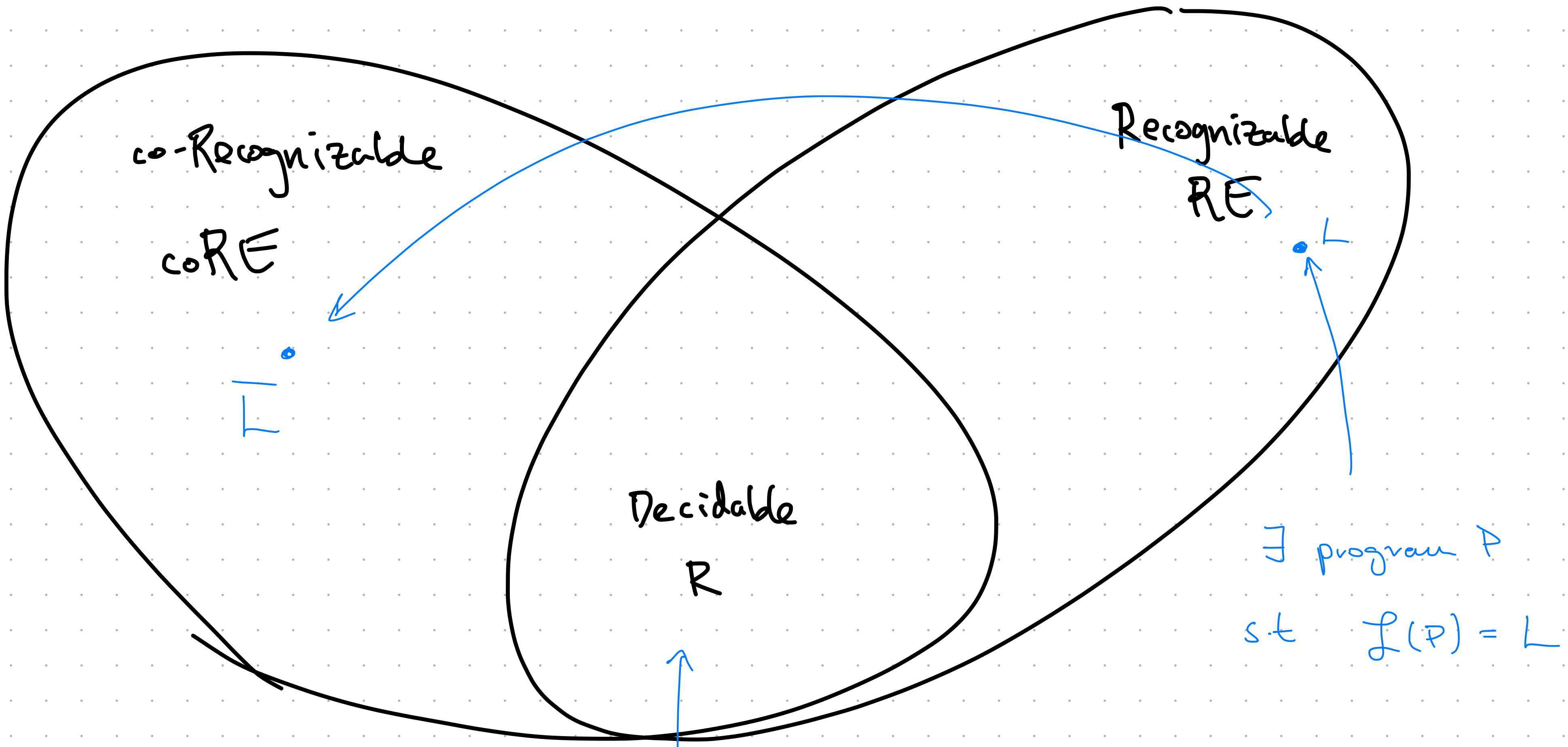
$\exists$ program P

s.t $\mathcal{L}(P) = L$.

Solvable in finite time

$\exists$ decider D s.t. $x \in L \Rightarrow D$ accepts x

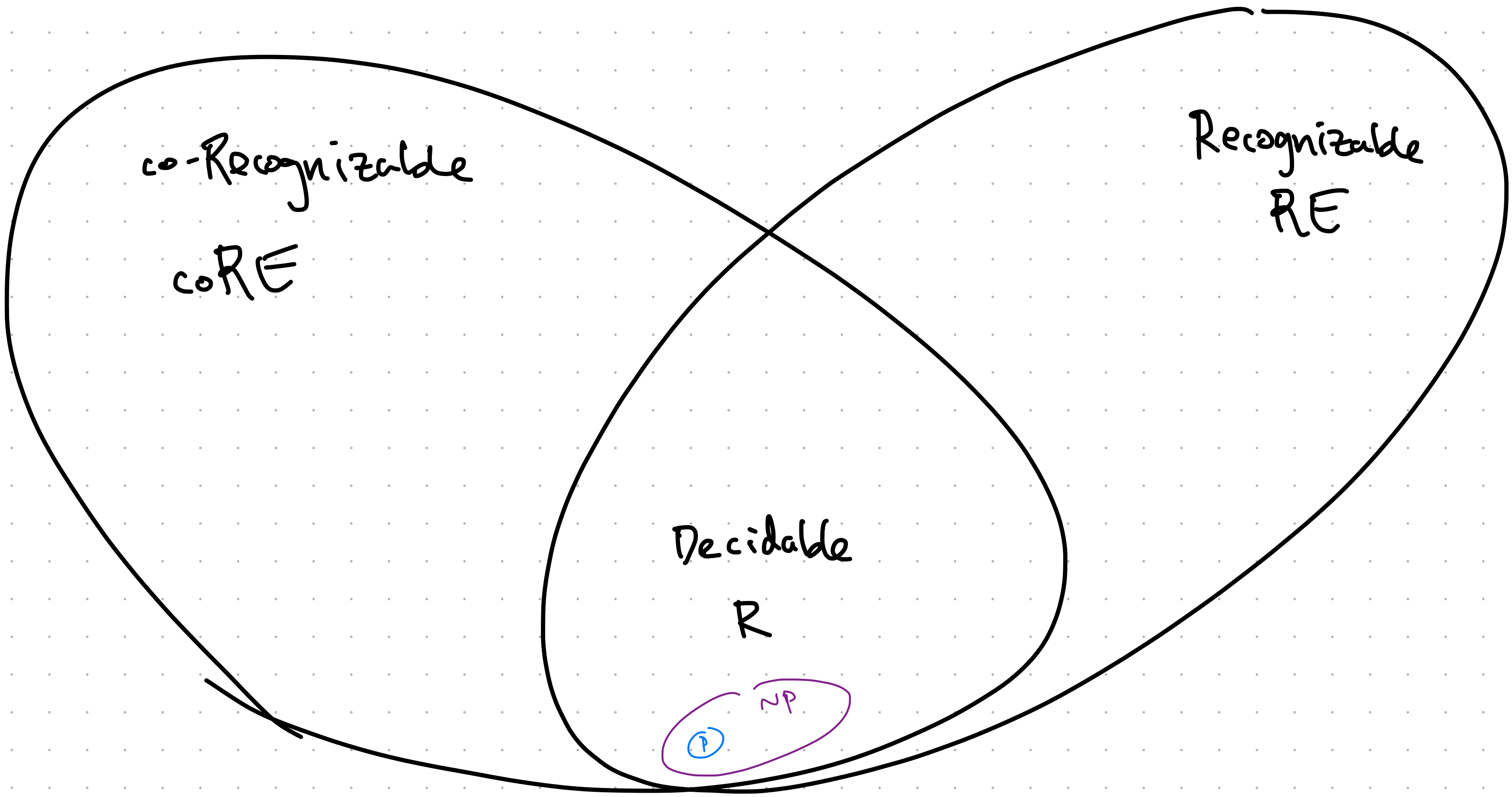
$x \notin L \Rightarrow D$ rejects x

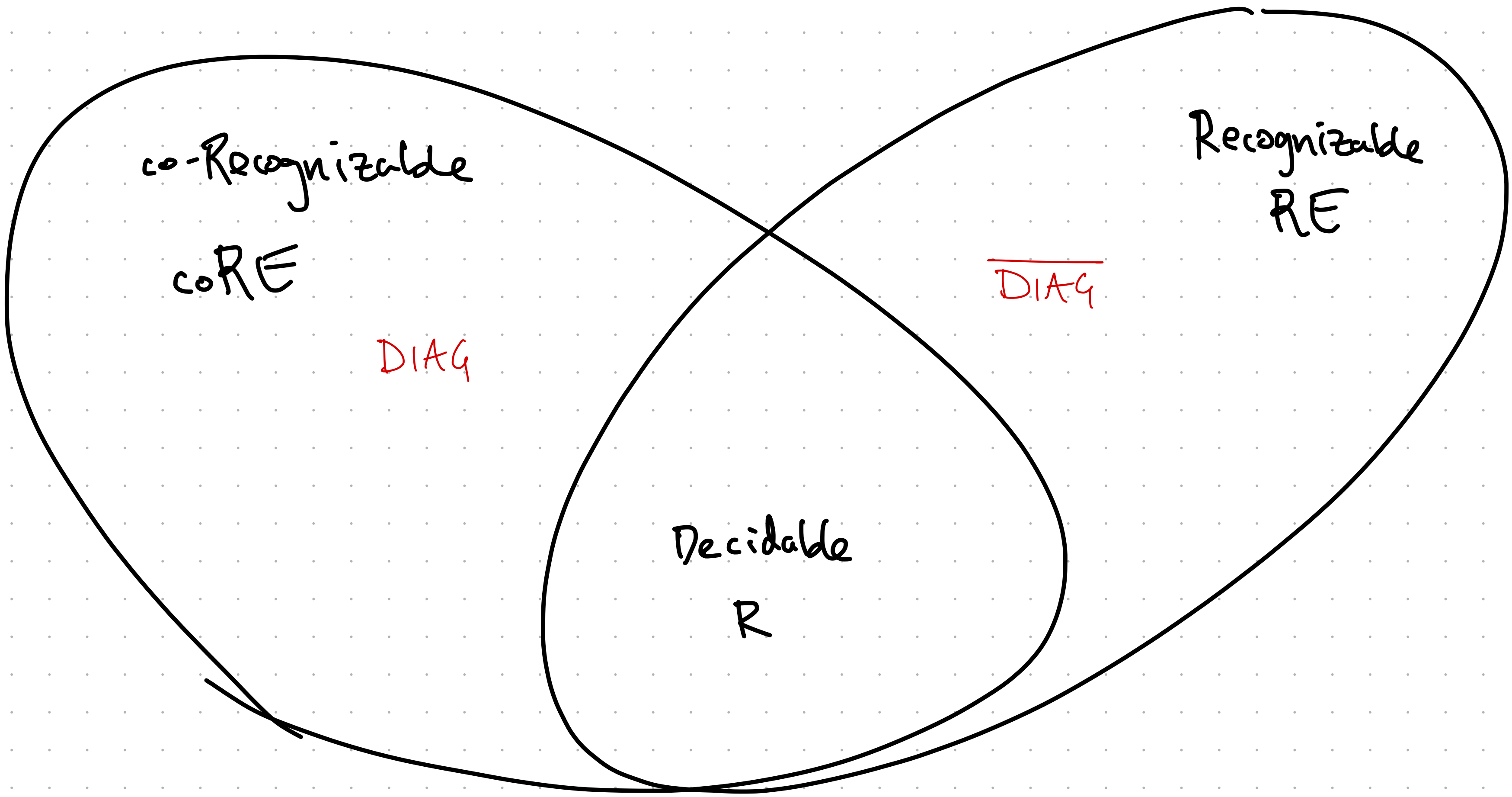


Solvable in finite time

$\exists$ decider D s.t. $x \in L \Rightarrow D$ accepts x

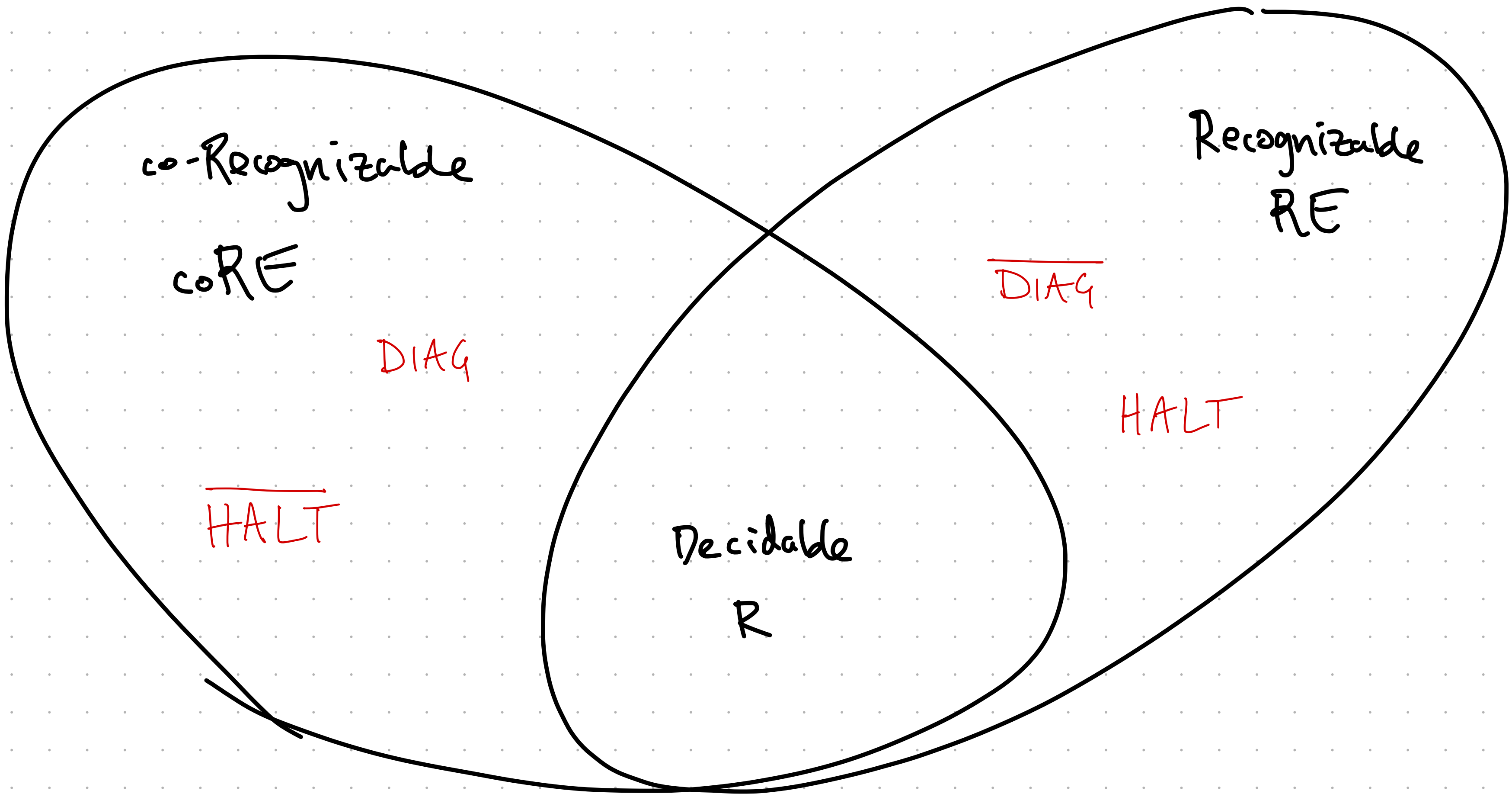
$x \notin L \Rightarrow D$ rejects x


co-Recognizable
coRE
Recognizable
RE
Decidable
R
NP
P



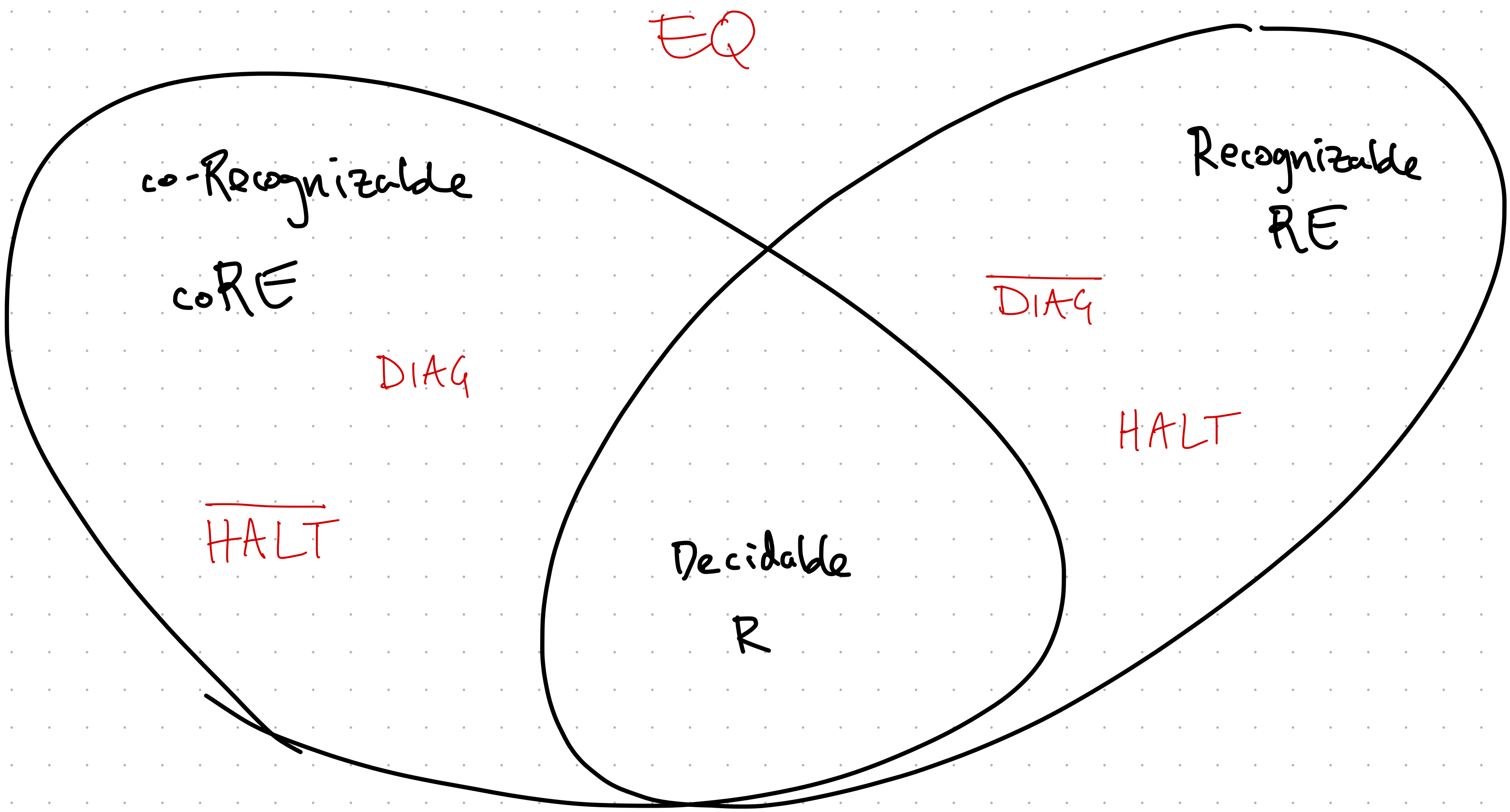
$$\text{DIAG} = \{ \langle P \rangle : P \text{ does not accept } \langle P \rangle \}$$

$$\overline{\text{DIAG}} = \{ \langle P \rangle : P \text{ accepts } \langle P \rangle \}$$



$$\text{HALT} = \{ \langle Q, x \rangle : Q \text{ halts on input } x \}$$

$$\overline{\text{HALT}} = \{ \langle Q, x \rangle : Q \text{ does not halt on input } x \}$$



$$EQ = \{ \langle P \rangle \# \langle Q \rangle : \mathcal{L}(P) = \mathcal{L}(Q) \}$$

Given two programs, do they recognize the same language?

Announcements

* HW8: Out next week
  - → Still due April 29 (skip next week)
  - → Regular length.

* All Prelim 2 regrades processed this weekend.

$$EQ = \{ \langle P \rangle \# \langle Q \rangle : \mathcal{L}(P) = \mathcal{L}(Q) \}$$

Given two programs, do they recognize the same language?

Theorem. $EQ \notin RE \cup coRE$.

↳ Not Recognizable, nor coRecognizable!

$$EQ = \{ \langle P \rangle \# \langle Q \rangle : \mathcal{L}(P) = \mathcal{L}(Q) \}$$

Given two programs, do they recognize the same language?

<u>Theorem</u>. $EQ \notin RE \cup coRE$.

↳ Not Recognizable, nor coRecognizable!

Proof Approach: Reduction from the Halting Problem.

Recall. HALT $\notin$ coRE

↳ Determining if Q halts on input x is recognizable, but not decidable.

<u>Recall</u>. HALT $\notin$ coRE

↳ Determining if Q halts on input x
is recognizable, but not decidable.

$$\underbrace{\text{HALT} \leq \text{EQ}}_{\text{Computable Reduction } R} \implies \text{EQ} \notin \text{coRE}.$$

Recall. HALT $\notin$ coRE

↳ Determining if Q halts on input x is recognizable, but not decidable.

$$\underbrace{\text{HALT} \leq \text{EQ}}_{\text{Computable Reduction } R} \implies \text{EQ} \notin \text{coRE}.$$

$$\langle Q, x \rangle \xrightarrow{R} \langle P_0, P_1 \rangle$$

Q halts on x $\implies$ $\mathcal{L}(P_0) = \mathcal{L}(P_1)$

Q does not halt on x $\implies$ $\mathcal{L}(P_0) \neq \mathcal{L}(P_1)$

Reduction from HALT to EQ.

Given $\langle Q, x \rangle$, write descriptions of P_0, P_1 as

$\underline{P_0}$

on input w

$\underline{P_1}$

on input w

Output $\langle P_0 \rangle \# \langle P_1 \rangle$

Reduction from HALT to EQ.

Given $\langle Q, x \rangle$, write descriptions of P_0, P_1 as

$\underline{P_0}$

on input w

ignore w

$\boxed{\text{Accept}}$

$\underline{P_1}$

on input w

Output $\langle P_0 \rangle \# \langle P_1 \rangle$

Reduction from HALT to EQ.

Given $\langle Q, x \rangle$, write descriptions of P_0, P_1 as

P_0

on input w
ignore w
$\boxed{\text{Accept}}$

P_1

on input w
ignore w, and run Q on x
$\boxed{\text{Accept}}$

// Hard code description of Q & x

Output $\langle P_0 \rangle \# \langle P_1 \rangle$

Reduction from HALT to EQ.

Given $\langle Q, x \rangle$, write descriptions of P_0, P_1 as

$\underline{P_0}$

on input w

ignore w

$\boxed{\text{Accept}}$

$\mathcal{L}(P_0) = \Sigma^*$

$\underline{P_1}$

on input w

ignore w, and run Q on x

$\boxed{\text{Accept}}$

$\mathcal{L}(P_1) = ?$

Output $\langle P_0 \rangle \# \langle P_1 \rangle$

P_1

on input w

ignore w, and run Q on x

$\boxed{\text{Accept}}$

$$\mathcal{L}(P_1) = \{ w \in \Sigma^* : P_1 \text{ accepts } w \}$$

$\langle Q, x \rangle \in HALT \Rightarrow Q$ halts on x in finite time

$\underline{P_1}$

on input w

ignore w, and run Q on x

$\boxed{\text{Accept}}$

$$\mathcal{L}(P_1) = \{ w \in \Sigma^* : P_1 \text{ accepts } w \}$$

$\langle Q, x \rangle \in HALT \implies Q$ halts on x in finite time

$\implies \forall w \in \Sigma^*,\ P_1$ accepts w

$\underline{P_1}$

on input w

ignore w, and run Q on x

$\boxed{\text{Accept}}$

$$\mathcal{L}(P_1) = \{ w \in \Sigma^* : P_1 \text{ accepts } w \}$$

$\langle Q, x \rangle \in \text{HALT} \implies Q$ halts on x in finite time

$\implies \forall w \in \Sigma^*$, P_1 accepts w

$\implies \mathcal{L}(P_1) = \Sigma^*$

P_1

on input w

ignore w, and run Q on x

$\boxed{\text{Accept}}$

NEVER GET TO $\boxed{\text{Accept}}$ instruction.

$$\mathcal{L}(P_1) = \{ w \in \Sigma^* : P_1 \text{ accepts } w \}$$

$\langle Q, x \rangle \in HALT \implies Q$ halts on x in finite time

$\implies \forall w \in \Sigma^*$, P_1 accepts w

$\implies \mathcal{L}(P_1) = \Sigma^*$

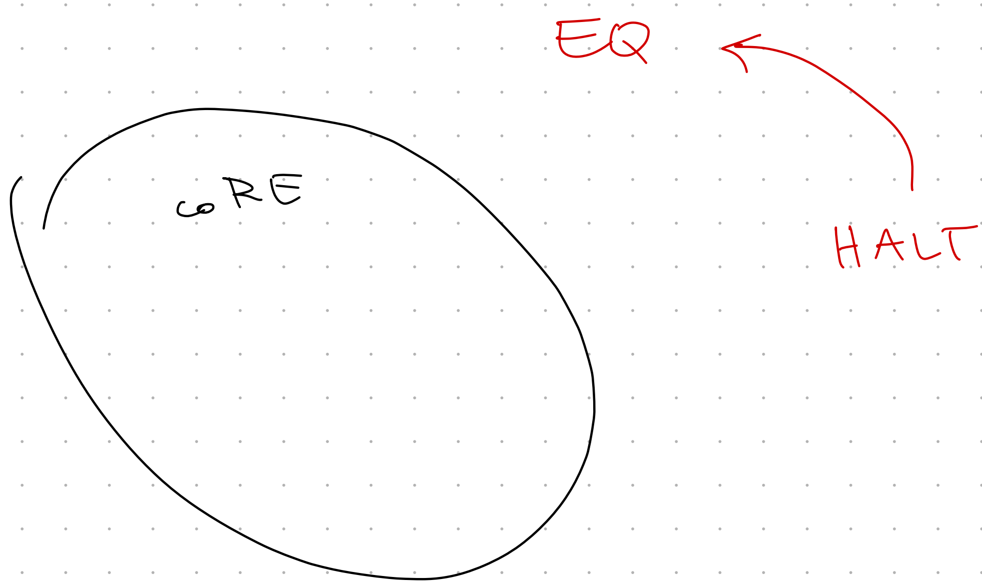
$\langle Q, x \rangle \notin HALT \implies Q$ does NOT halt on x

$\implies \forall w \in \Sigma^*$, P_1 does not halt on w

$\implies \mathcal{L}(P_1) = \emptyset$

HALT $\longrightarrow$ EQ.

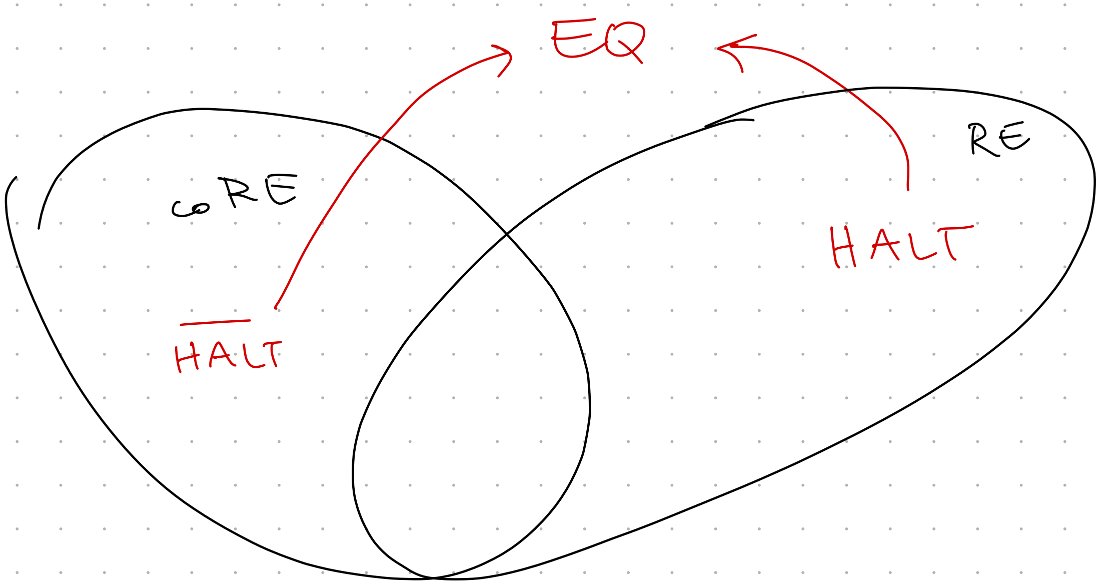
Q halts on x $\iff$ $\mathcal{L}(P_0) = \mathcal{L}(P_1) = \Sigma^*$



$\Longrightarrow$ EQ $\notin$ coRE

$\overline{HALT} \longrightarrow EQ.$

Q does not halt on x $\iff$ $\mathcal{L}(P_0) = \mathcal{L}(P_1)$



$\Rightarrow EQ \notin RE$

Recall: $\overline{HALT} \notin RE$.

So $\overline{HALT} \leq EQ \implies EQ \notin RE$

Computable Reduction

$$\langle Q, x \rangle \xrightarrow{\quad R \quad} \langle P_0 \rangle \# \langle P_1 \rangle$$

Q does NOT halt on x $\implies$ $\mathcal{L}(P_0) = \mathcal{L}(P_1)$

Q halts on x $\implies$ $\mathcal{L}(P_0) \neq \mathcal{L}(P_1)$

Reduction from $\overline{HALT}$ to $\overline{EQ}$.

Given $\langle Q, x \rangle$, write descriptions of P_0, P_1 as

$\underline{P_0}$

on input w

$\underline{P_1}$

on input w

Output $\langle P_0 \rangle \# \langle P_1 \rangle$

Reduction from $\overline{HALT}$ to EQ.

Given $\langle Q, x \rangle$, write descriptions of P_0, P_1 as

P_0

on input w
ignore w
$\boxed{\text{Reject}}$

$\mathcal{L}(P_0) = \emptyset$

P_1

on input w
ignore w, and run Q on x
$\boxed{\text{Accept}}$

$\mathcal{L}(P_1) = ?$

Output $\langle P_0 \rangle \# \langle P_1 \rangle$

$\underline{P_1}$

on input w

ignore w, and run Q on x

Accept

$$\langle Q, x \rangle \in \overline{HALT} \implies Q \text{ does not halt on } x$$

$$\implies \mathcal{L}(P_1) = \emptyset$$

$$\langle Q, x \rangle \in HALT \implies Q \text{ halts on } x$$

$$\implies \mathcal{L}(P_1) = \Sigma^*$$

$$\mathcal{L}(P_0) = \mathcal{L}(P_1) = \emptyset \iff Q \text{ does not halt on } x$$

$$EQ = \{ \langle P_0 \rangle \# \langle P_1 \rangle \; : \; \mathcal{L}(P_0) = \mathcal{L}(P_1) \}$$

$$\boxed{EQ \notin RE \cup coRE}$$

No way to <u>prove</u> or <u>refute</u>

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.

Languages about Programs

* Examples

  DIAG, HALT, EQ, ...

* Many such languages are undecidable.

Do we have to show a new reduction for each one?

Semantic Languages

* A language of Program encodings L is semantic if

  $\langle P \rangle \in L$ is determined by $\underbrace{\text{the language } \mathcal{L}(P)}_{\text{input-output behavior of } P}$.

Semantic Languages

* A language of Program encodings L is semantic if

$\langle P \rangle \in L$ is determined by the language $\mathcal{L}(P)$.

Example Semantic Languages

$$L_{EMPTY} = \{ \langle P \rangle : \mathcal{L}(P) = \emptyset \}$$

$$L_{ALL} = \{ \langle P \rangle : \mathcal{L}(P) = \Sigma^* \}$$

$$L_{finite} = \{ \langle P \rangle : |\mathcal{L}(P)| \text{ is finite} \}$$

Non-Semantic Language

$$L_{for\text{-}loop} = \{ \langle P \rangle : P \text{ uses a for loop} \}$$

implementation detail

Rice's Theorem

$L \notin \{\emptyset, \Sigma^*\}$

Suppose L is a nontrivial semantic language.

Then, L is undecidable.

Implication for Program Analysis / Security.

Rice's Theorem $\Longrightarrow$ Impossible to prove correctness / security of arbitrary programs.

<u>Proof</u>. Reduction from HALT.

* L is non trivial $\Longrightarrow \exists$ some $\langle M \rangle \in L$

Suppose $\langle$ on input w, $\boxed{\text{Reject}}$ $\rangle \notin L$ (similar argument for case $\notin L$)

<u>Reduction</u>

Given $\langle Q, x \rangle$, construct P as

<u>P</u>
on input w

Output $\langle P \rangle$

Proof. Reduction from HALT.

* L is non trivial $\Longrightarrow$ $\exists$ some $\langle M \rangle \in L$

Suppose $\langle$ on input w, $\boxed{\text{Reject}}$ $\rangle \notin L$ (similar argument for case $\notin L$)

Reduction

Given $\langle Q, x \rangle$, construct P as

P
on input w
run Q on x
run M on w
if M accepts, $\boxed{\text{Accept}}$

Output $\langle P \rangle$

Proof. Reduction from HALT.

* L is non trivial $\Longrightarrow \exists$ some $\langle M \rangle \in L$

Suppose $\langle$ on input w, $\boxed{\text{Reject}}$ $\rangle \notin L$ (similar argument for case $\notin L$)

Reduction

Given $\langle Q, x \rangle$, construct P as

P
on input w
run Q on x
run M on w
if M accepts, $\boxed{\text{Accept}}$

$\mathcal{L}(P) = \mathcal{L}(M) \Longrightarrow \langle P \rangle \in L$

$\Longleftrightarrow$

$\langle Q, x \rangle \in \text{HALT}$

Output $\langle P \rangle$

Proof. Reduction from HALT.

$\ast$ L is non trivial $\Longrightarrow$ $\exists$ some $\langle M \rangle \in L$

Suppose $\langle$ on input w, $\boxed{\text{Reject}}$ $\rangle \notin L$ (similar argument for case $\notin L$)

Reduction

Given $\langle Q, x \rangle$, construct P as

P
on input w
run Q on x
run M on w
if M accepts, $\boxed{\text{Accept}}$

$\mathcal{L}(P) = \mathcal{L}(M) \Rightarrow \langle P \rangle \in L$

$\Longleftrightarrow$

$\langle Q, x \rangle \in HALT$

$\mathcal{L}(P) = \emptyset \Rightarrow \langle P \rangle \notin L$

$\Longleftrightarrow$

$\langle Q, x \rangle \notin HALT$

Output $\langle P \rangle$