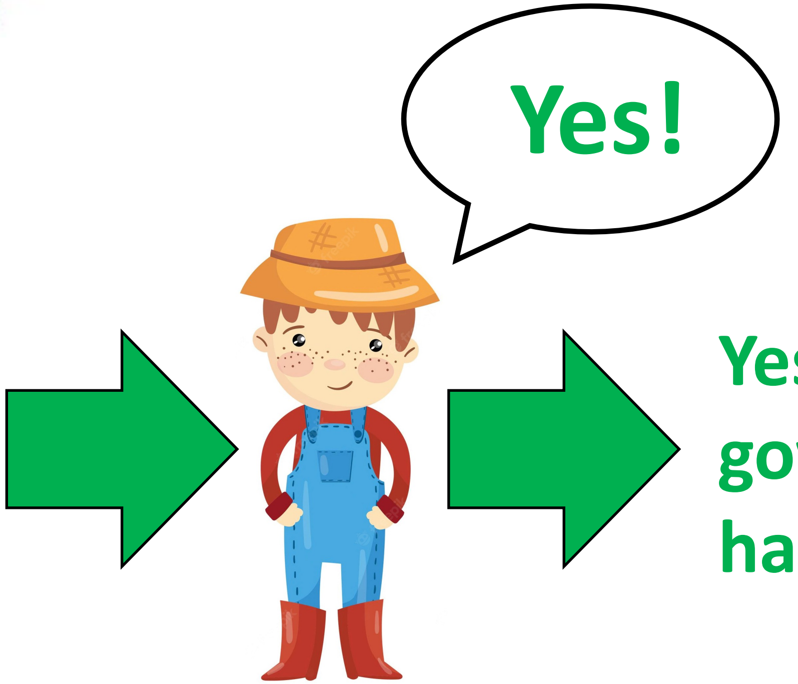
Undecidability

CSCI 338

Question: Does the government have aliens?

Bread Recipe:

1. Add flour, water, sugar, yeast, salt.
2. If the government has aliens, add an egg.
3. Mix, bake, eat.



Yes, the government has aliens.

Frank

Can always perfectly answer the question: "Does a recipe use eggs?"

Does the government have aliens?:

1. Write down the following recipe:

   Bread Recipe:
   1. Add flour, water, sugar, yeast, salt.
   2. If the government has alien, add an egg.
   3. Mix, bake, eat.

2. Ask Frank if the recipe uses eggs.
3. If Frank says yes, <u>yes, they got aliens</u>. If Frank says no, <u>no, they don't</u>.

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. Run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

Does the government have aliens?:

1. Write down the following recipe:

> Bread Recipe:
> 1. Add flour, water, sugar, yeast, salt.
> 2. If the government has alien, add an egg.
> 3. Mix, bake, eat.

2. Ask Frank if the recipe uses eggs.
3. If Frank says yes, yes, they got aliens. If Frank says no, no, they don't.

A_{TM} Decider (does N accept ω?):

1. Construct TM M_2 on input $\langle x \rangle$:
    1. Run N on ω and accept if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

Question?

Does the government have aliens?:

1. Write down the following recipe:

   <u>Bread Recipe</u>:
   1. Add flour, water, sugar, yeast, salt.
   2. If the government has alien, add an egg.
   3. Mix, bake, eat.

2. Ask Frank if the recipe uses eggs.
3. If Frank says yes, <u>yes, they got aliens</u>. If Frank says no, <u>no, they don't</u>.

A_{TM} Decider (does N accept ω?):

1. Construct TM M_2 on input $\langle x \rangle$:
   1. Run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

Question?

Does the government have aliens?:

1. Write down the following recipe:

   Bread Recipe:
   1. Add flour, water, sugar, yeast, salt.
   2. If the government has alien, add an egg.
   3. Mix, bake, eat.

2. **Ask Frank if the recipe uses eggs.**
3. If Frank says yes, yes, they got aliens. If Frank says no, no, they don't.

A_{TM} Decider (does N accept ω?):

1. Construct TM M_2 on input $\langle x \rangle$:
   1. Run N on ω and accept if N does.
2. **Run H on $\langle M_2 \rangle$.**
3. If H accepts, reject. If H rejects, accept.

Available Tool

Question?

Does the government have aliens?:

1. Write down the following recipe:

   Bread Recipe:
   1. Add flour, water, sugar, yeast, salt.
   2. If the government has alien, add an egg.
   3. Mix, bake, eat.
2. **Ask Frank if the recipe uses eggs.**
3. If Frank says yes, yes, they got aliens. If Frank says no, no, they don't.

A_{TM} Decider (does N accept ω?):

1. Construct TM M_2 on input $\langle x \rangle$:
   1. Run N on ω and accept if N does.
2. **Run H on $\langle M_2 \rangle$.**
3. If H accepts, reject. If H rejects, accept.

Available Tool

Recipe uses egg $\Leftrightarrow$ Government has aliens

$L(M_2) \neq \emptyset \Leftrightarrow N$ **accepts** ω

Recipe/algorithm/TM that forces the tool to answer the question.

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. Run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

TODO: Argue that S is a decider for A_{TM}

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. Run N on ω and accept if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

If N accepts ω, … S will accept. If N does not accept ω, … S will reject.

E_{TM}

Claim: $E_{TM} = \{\langle M\rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_2 on input $\langle x\rangle$:
   1. Run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2\rangle$.
3. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

If N accepts ω, $L(M_2) = \Sigma^*$, …

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. Run N on ω and accept if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, reject. If H rejects, accept.

If N accepts ω, $L(M_2) = \Sigma^*$, H will reject, …

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
   1. Run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

If N accepts ω, $L(M_2) = \Sigma^*$, H will reject, and S will accept.

E_{TM}

Claim: $E_{TM} = \{\langle M\rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_2 on input $\langle x\rangle$:
   1. Run N on ω and accept if N does.
2. Run H on $\langle M_2\rangle$.
3. If H accepts, reject. If H rejects, accept.

If N accepts ω, $L(M_2) = \Sigma^*$, H will reject, and S will accept. If N does not accept ω, $L(M_2) = \emptyset$, H will accept, and S will reject.

E_{TM}

Claim: $E_{TM} = \{\langle M \rangle : M$ is a TM and $L(M) = \emptyset\}$ is undecidable.

Proof: Suppose E_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega \rangle$

1. Construct TM M_2 on input $\langle x \rangle$:
    1. Run N on ω and <u>accept</u> if N does.
2. Run H on $\langle M_2 \rangle$.
3. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

If N accepts ω, $L(M_2) = \Sigma^*$, H will reject, and S will accept. If N does not accept ω, $L(M_2) = \emptyset$, H will accept, and S will reject. Thus, S decides A_{TM}, which is a contradiction, so E_{TM} is undecidable.

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof:

?

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle: A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. ???

To show EQ_{TM} is undecidable, use it to decide A_{TM}.

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle: A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. ???

We have a way (H) to test if two TMs have the same language.

How could we use that to test if a TM's accepts some input?

Plan: ?

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. ???

We have a way (H) to test if two TMs have the same language.

How could we use that to test if a TM's accepts some input?

Plan: Make two TMs that have the same language if and only if N accepts ω.

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
    1. accept.

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. accept.

$\boldsymbol{L(M_2) = ?}$

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
    1. accept.

$\boldsymbol{L(M_2) = \Sigma^*}$

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle: A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
    1. accept.
2. Construct TM M_2 on input $\langle y\rangle$:
    1. ?

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. accept.
2. Construct TM M_2 on input $\langle y\rangle$:
   1. Run N on ω and accept if N does.

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. <u>accept.</u>
2. Construct TM M_2 on input $\langle y\rangle$:
   1. Run N on ω and <u>accept</u> if N does.

$\boldsymbol{N}$ **accepts** $\boldsymbol{\omega}$

$$\Updownarrow$$

$$\boldsymbol{L(M_2) = \Sigma^*}$$

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. accept.
2. Construct TM M_2 on input $\langle y\rangle$:
   1. Run N on ω and accept if N does.
3. Run H on $\langle M_1, M_2\rangle$.
4. If H accepts, ???

$$\boxed{\begin{array}{c} \boldsymbol{N} \textbf{ accepts } \boldsymbol{\omega} \\ \Updownarrow \\ \boldsymbol{L(M_2) = \Sigma^*} \end{array}}$$

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. accept.
2. Construct TM M_2 on input $\langle y\rangle$:
   1. Run N on ω and accept if N does.
3. Run H on $\langle M_1, M_2\rangle$.
4. If H accepts, accept. If H rejects, reject.

$\boldsymbol{N}$ **accepts** $\boldsymbol{\omega}$

$\Updownarrow$

$\boldsymbol{L(M_2) = \Sigma^*}$

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. accept.
2. Construct TM M_2 on input $\langle y\rangle$:
   1. Run N on ω and accept if N does.
3. Run H on $\langle M_1, M_2\rangle$.
4. If H accepts, accept. If H rejects, reject.

$\boldsymbol{N}$ **accepts** $\boldsymbol{\omega}$

$\Updownarrow$

$\boldsymbol{L(M_2) = \Sigma^*}$

If N accepts ω, then M_1 and M_2 have the same language (Σ^*) and S accepts. If N does not accept ω, then they have different languages and S rejects. Thus, S decides A_{TM}, which is a contradiction, so EQ_{TM} is undecidable.

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. <u>accept.</u>
2. Construct TM M_2 on input $\langle y\rangle$:
   1. Run N on ω and <u>accept</u> if N does.
3. Run H on $\langle M_1, M_2\rangle$.
4. If H accepts, <u>accept</u>. If H rejects, <u>reject</u>.

$$\boldsymbol{N \text{ accepts } \omega} \\ \Updownarrow \\ \boldsymbol{L(M_2) = \Sigma^*}$$

Lots of ways to write the recipe!

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. accept.
2. Construct TM M_2 on input $\langle y\rangle$:
   1. Run N on ω and accept if N does.
3. Run H on $\langle M_1, M_2\rangle$.
4. If H accepts, accept. If H rejects, reject.

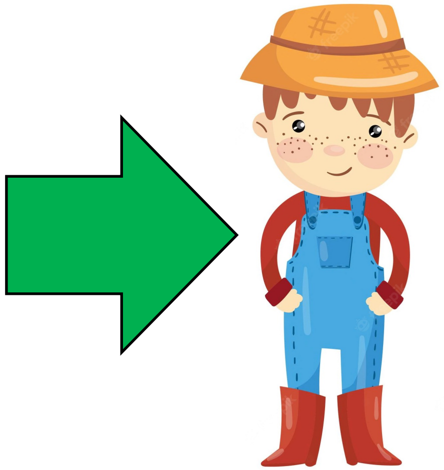
$$\boldsymbol{N} \textbf{ accepts } \boldsymbol{\omega} \;\Updownarrow\; \boldsymbol{L(M_1) = L(M_2)}$$

Lots of ways to write the recipe!

Question: Does the government have aliens?

<u>Bread Recipe</u>:

1. Add flour, water, sugar, yeast, salt.
2. If government has aliens, add an egg.
3. Mix, bake, eat.

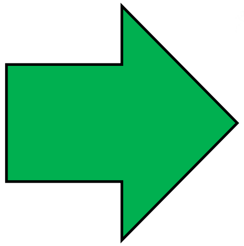


Question: Does the government have aliens?

<u>Bread Recipe</u>:
1. Add flour, water, sugar, yeast, salt.
2. If government has aliens, add an egg.
3. Mix, bake, eat.

<u>Bread Recipe</u>:
1. Add flour, water, sugar, yeast, salt.
2. If government doesn’t have aliens, add an egg.
3. Mix, bake, eat.

<u>Recipe</u>:
1. If government has aliens, add an egg.



Question: Does the government have aliens?

Bread Recipe:

1. Add flour, water, sugar, yeast, salt.
2. If government has aliens, add an egg.
3. Mix, bake, eat.

Bread Recipe:

1. Add flour, water, sugar, yeast, salt.
2. If government doesn’t have aliens, add an egg.
3. Mix, bake, eat.

Recipe:

1. If government has aliens, add an egg.

Won’t Work Recipe:

1. If government has aliens, add flour.
2. Add eggs.

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. accept.
2. Construct TM M_2 on input $\langle y\rangle$:
   1. Run N on ω and accept if N does.
3. Run H on $\langle M_1, M_2\rangle$.
4. If H accepts, accept. If H rejects, reject.

$\boldsymbol{N}$ **accepts** $\boldsymbol{\omega}$

$\Updownarrow$

$\boldsymbol{L(M_1) = L(M_2)}$

Lots of ways to write the recipe!

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
   1. <u>reject.</u>
2. Construct TM M_2 on input $\langle y\rangle$:
   1. Run N on ω and <u>accept</u> if N does.
3. Run H on $\langle M_1, M_2\rangle$.
4. If H accepts, <u>**reject**</u>. If H rejects, <u>**accept**</u>.

$$\boldsymbol{N} \textbf{ accepts } \boldsymbol{\omega} \Updownarrow \boldsymbol{L(M_1) = L(M_2)}$$

Lots of ways to write the recipe!

EQ_{TM}

Claim: $EQ_{TM} = \{\langle A, B\rangle : A, B$ are TMs and $L(A) = L(B)\}$ is undecidable.

Proof: Suppose EQ_{TM} is decidable and let TM H be its decider.

Build a TM S that decides A_{TM}:

S = on input $\langle N, \omega\rangle$

1. Construct TM M_1 on input $\langle x\rangle$:
    1. <u>accept.</u>
2. Construct TM M_2 on input $\langle y\rangle$:
    1. Run N on ω.
    2. If N rejects, <u>reject.</u>
    3. <u>accept.</u>
3. Run H on $\langle M_1, M_2\rangle$.
4. If H accepts, <u>accept</u>. If H rejects, <u>reject</u>.

$$\boldsymbol{N \text{ accepts } \omega} \\ \Updownarrow \\ \boldsymbol{L(M_1) = L(M_2)}$$

Lots of ways to write the recipe!