Review
CSCI 338
Claim: $E_{TM} = \{ \langle M \rangle : M \text{ is a TM and } L(M) = \emptyset \}$ is undecidable.

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

Build a TM $S$ that decides $A_{TM}$:

$S = \text{on input } \langle N, \omega \rangle$

1. Construct TM $M_2$ on input $\langle x \rangle$:
   1. Run $N$ on $\omega$ and accept if $N$ does.
   2. Run $H$ on $\langle M_2 \rangle$.
2. If $H$ accepts, reject. If $H$ rejects, accept.
Question: Does the government have aliens?
Question: Does the government have aliens?

Frank

Can always perfectly answer the question: “Does a recipe use eggs?”
Question: Does the government have aliens?

Frank

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

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

Frank
Can always perfectly answer the question: “Does a recipe use eggs?”
Question: Does the government have aliens?

Yes!

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

Frank
Can always perfectly answer the question:
“Does a recipe use eggs?”
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.

Frank
Can always perfectly answer the question: “Does a recipe use eggs?”
Question: Does the government have aliens?

Frank can always perfectly answer the question: “Does a recipe use eggs?”

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

Yes!
Yes, the government has aliens.
Question: Does the government have aliens?

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

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

Frank can always perfectly answer the question: “Does a recipe use eggs?”

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

No! No, the government doesn’t have aliens.
Does the government have aliens?:
Does the government have aliens?:
1. Write down the following recipe:

<table>
<thead>
<tr>
<th>Bread Recipe:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Add flour, water, sugar, yeast, salt.</td>
</tr>
<tr>
<td>2. If the government has aliens, add an egg.</td>
</tr>
<tr>
<td>3. Mix, bake, eat.</td>
</tr>
</tbody>
</table>
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 aliens, add an egg.
   3. Mix, bake, eat.

2. ???
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 aliens, add an egg.
  3. Mix, bake, eat.

  2. Ask Frank if the recipe uses eggs.
Does the government have aliens?:

1. Write down the following recipe:

<table>
<thead>
<tr>
<th>Bread Recipe:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Add flour, water, sugar, yeast, salt.</td>
</tr>
<tr>
<td>2. If the government has aliens, add an egg.</td>
</tr>
<tr>
<td>3. Mix, bake, eat.</td>
</tr>
</tbody>
</table>

2. Ask Frank if the recipe uses eggs.
3. If Frank says yes, ???
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 aliens, add an egg.
   3. Mix, bake, eat.

2. Ask Frank if the recipe uses eggs.

3. If Frank says yes, *yes, the government has aliens*. If Frank says no, *no, the government does not have aliens.*
Claim: $E_T^M = \{\langle M \rangle : M \text{ is a TM and } L(M) = \emptyset \}$ is undecidable.

Proof: Suppose $E_T^M$ is decidable and let TM $H$ be its decider.

Build a TM $S$ that decides $A_T^M$:

1. Construct TM $M_2$ on input $\langle x \rangle$:
   1. Run $N$ on $\omega$ and accept if $N$ does.
   2. Run $H$ on $\langle M_2 \rangle$.
   3. If $H$ accepts, reject. If $H$ rejects, accept.
Claim: \( E_{TM} = \{\langle M \rangle: M \text{ is a TM and } L(M) = \emptyset \} \) is undecidable.

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

Build a TM \( S \) that decides \( A_{TM} \):

\[ S = \text{on input } \langle N, \omega \rangle \]

1. Construct TM \( M_2 \) on input \( \langle x \rangle \):
   1. Run \( N \) on \( \omega \) and accept if \( N \) does.
   2. Run \( H \) on \( \langle M_2 \rangle \).

3. If \( H \) accepts, reject. If \( H \) rejects, accept.
Claim: $E_{TM} = \{\langle M \rangle : M \text{ is a TM and } L(M) = \emptyset \}$ 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 $\omega$ and accept if $N$ does.
   2. Run $H$ on $\langle M_2 \rangle$.
2. If $H$ accepts, reject. If $H$ rejects, accept.
Claim: \( E_{TM} = \{\langle M\rangle: M \text{ is a TM and } L(M) = \emptyset\} \) is undecidable.

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

Build a TM \( S \) that decides \( A_{TM} \):

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

The question I care about (aliens?)
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 $A_{TM}$:

$S = \text{on input } \langle N, \omega \rangle$

1. Construct TM $M_2$ on input $\langle x \rangle$:
   1. Run $N$ on $\omega$ and accept if $N$ does.
   2. Run $H$ on $\langle M_2 \rangle$.
   3. If $H$ accepts, reject. If $H$ rejects, accept.
Claim: \(FINITE_{TM} = \{\langle M \rangle: M \text{ is a TM and accepts a finite number of strings} \}\) is undecidable.

Proof: Suppose \(FINITE_{TM}\) is decidable and let TM \(H\) be its decider.

Build a TM \(S\) that decides \(A_{TM}\):

\[
S = \text{ on input } \langle N, \omega \rangle \\
1. \text{ Construct TM } M_2 \text{ on input } \langle x \rangle:
   \begin{align*}
   1. & \text{ Run } N \text{ on } \omega \text{ and accept if } N \text{ does.} \\
   2. & \text{ Run } H \text{ on } \langle M_2 \rangle. \\
   3. & \text{ If } H \text{ accepts, reject. If } H \text{ rejects, accept.}
   \end{align*}
\]

If \(N\) accepts \(\omega\), \(L(M_2) = \Sigma^*\) (infinite) and \(S\) accepts. If \(N\) does not accept \(\omega\), \(L(M_2) = \emptyset\) (finite) and \(S\) rejects. Therefore, \(S\) is a decider for \(A_{TM}\), which is a contradiction, so \(FINITE_{TM}\) is undecidable.
Claim: $EVEN_{TM} = \{\langle M \rangle: M$ is a TM and $L(M)$ contains all even length strings$\}$ is undecidable.

Proof: Suppose $EVEN_{TM}$ 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 $\omega$ and accept if $N$ does.
   2. Run $H$ on $\langle M_2 \rangle$.
   3. If $H$ accepts, accept. If $H$ rejects, reject.

If $N$ accepts $\omega$, $L(M_2) = \Sigma^*$ (contains all even length strings) and $S$ accepts.
If $N$ does not accept $\omega$, $L(M_2) = \emptyset$ (does not contain all even length strings) and $S$ rejects. Thus, $S$ is a decider for $A_{TM}$, which is a contradiction, so $EVEN_{TM}$ is undecidable.
Claim: $EVEN_{TM} = \{\langle M \rangle: M \text{ is a TM and } L(M) \text{ contains all even length strings} \}$ is undecidable.

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

Build a TM $S$ that decides $A_{TM}$:

$S = \text{on input } \langle N, \omega \rangle$

1. Construct TM $M_2$ on input $\langle x \rangle$:
   
   1. If $x$ has odd length, reject.
   
   2. If $x$ has even length, run $N$ on $\omega$.

   3. If $N$ accepts $\omega$, accept. If $N$ rejects $\omega$, reject.

2. Run $H$ on $\langle M_2 \rangle$. 

$L(M_2)$ contains all even length strings

$\uparrow$

$N$ accepts $\omega$
Claim: \(EVEN_{TM} = \{\langle M \rangle : M \text{ is a TM and } L(M) \text{ contains all even length strings}\}\) is undecidable.

Proof: Suppose \(EVEN_{TM}\) 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. If \(x\) has odd length, reject. \(\text{accept} \)
   2. If \(x\) has even length, run \(N\) on \(\omega\).
   3. If \(N\) accepts \(\omega\), accept. If \(N\) rejects \(\omega\), reject.
2. Run \(H\) on \(\langle M_2 \rangle\).

\(L(M_2)\) contains all even length strings
\(\uparrow\)
\(N\) accepts \(\omega\)
Claim: $EVEN_{TM} = \{\langle M \rangle: M \text{ is a TM and } L(M) \text{ contains all even length strings} \}$ is undecidable.

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

Build a TM $S$ that decides $A_{TM}$:

$S = \text{on input } \langle N, \omega \rangle$

1. Construct TM $M_2$ on input $\langle x \rangle$:
   1. If $x$ has odd length, reject.
   2. If $x$ has even length, run $N$ on $\omega$.
   3. If $N$ accepts $\omega$, accept. If $N$ rejects $\omega$, reject.

2. Run $H$ on $\langle M_2 \rangle$. 

$L(M_2)$ contains all even length strings

$\Downarrow$

$N$ accepts $\omega$
Claim: \( EVEN_{TM} = \{ \langle M \rangle : M \text{ is a TM and } L(M) \text{ contains all even length strings} \} \) is undecidable.

Proof: Suppose \( EVEN_{TM} \) is decidable and let TM \( H \) be its decider. Build a TM \( S \) that decides \( A_{TM} \):

\[
S = \text{on input } \langle N, \omega \rangle \\
1. \text{Construct TM } M_2 \text{ on input } \langle x \rangle : \\
   1. \text{If } x \text{ has odd length, reject.} \\
   2. \text{If } x \text{ has even length, run } N \text{ on } \omega. \\
   3. \text{If } N \text{ accepts } \omega, \text{ accept. If } N \text{ rejects } \omega, \text{ reject.} \\
2. \text{Run } H \text{ on } \langle M_2 \rangle.
\]
Claim: $HALT - \varepsilon_{TM} = \{\langle M \rangle: M \text{ is a TM and halts on empty input (i.e. } \varepsilon)\}$ is undecidable.

Proof: Suppose $HALT - \varepsilon_{TM}$ is decidable and let TM $H$ be its decider.

Build a TM $S$ that decides $A_{TM}$:

$S = \text{on input } \langle N, \omega \rangle$

1. Construct TM $M_2$ on input $\langle x \rangle$:
   1. Run $N$ on $\omega$ and accept if $N$ does.

2. Run $H$ on $\langle M_2 \rangle$.

3. If $H$ accepts, accept. If $H$ rejects, reject.

If $N$ accepts $\omega$, $M_2$ halts on $\varepsilon$ and $S$ accepts. If $N$ does not accept $\omega$, $M_2$ does not halt on $\varepsilon$ and $S$ rejects. Therefore, $S$ is a decider for $A_{TM}$, which is a contradiction, so $HALT - \varepsilon_{TM}$ is undecidable.
Claim: $HALT - \varepsilon_{TM} = \{\langle M \rangle: M \text{ is a TM and halts on empty input (i.e. } \varepsilon)\}$ is undecidable.

Proof: Suppose $HALT - \varepsilon_{TM}$ is decidable and let TM $H$ be its decider.

Build a TM $S$ that decides $A_{TM}$:

\[ S = \text{on input } \langle N, \omega \rangle \]

1. Construct TM $M_2$ on input $\langle x \rangle$:
   
   1. Run $N$ on $\omega$ and accept if $N$ accepts.
   2. Run $H$ on $\langle M_2 \rangle$.

$M_2$ halts on $\varepsilon$ \iff $N$ accepts $\omega$
Claim: $\text{HALT} - \varepsilon_{TM} = \{\langle M \rangle: M \text{ is a TM and halts on empty input (i.e. } \varepsilon)\}$ is undecidable.

Proof: Suppose $\text{HALT} - \varepsilon_{TM}$ is decidable and let TM $H$ be its decider.

Build a TM $S$ that decides $A_{TM}$:

$S = \text{on input } \langle N, \omega \rangle$

1. Construct TM $M_2$ on input $\langle x \rangle$:
   1. If $x \neq \varepsilon$, reject.
   2. If $x = \varepsilon$, run $N$ on $\omega$ and accept if $N$ accepts.

2. Run $H$ on $\langle M_2 \rangle$. 

$M_2$ halts on $\varepsilon$ ⇐ $N$ accepts $\omega$
Claim: $\text{HALT} - \varepsilon_{TM} = \{ \langle M \rangle : M \text{ is a TM and halts on empty input (i.e. } \varepsilon) \}$ is undecidable.

Proof: Suppose $\text{HALT} - \varepsilon_{TM}$ is decidable and let TM $H$ be its decider.

Build a TM $S$ that decides $A_{TM}$:

$S = \text{on input } \langle N, \omega \rangle$

1. Construct TM $M_2$ on input $\langle x \rangle$:
   1. If $x \neq \varepsilon$, reject. **accept**
   2. If $x = \varepsilon$, run $N$ on $\omega$ and **accept** if $N$ accepts.

2. Run $H$ on $\langle M_2 \rangle$.  

$M_2$ halts on $\varepsilon$ \iff $N$ accepts $\omega$
Claim: $HALT - \varepsilon_{TM} = \{\langle M \rangle: M \text{ is a TM and halts on empty input (i.e. } \varepsilon)\}$ is undecidable.

Proof: Suppose $HALT - \varepsilon_{TM}$ is decidable and let TM $H$ be its decider.

Build a TM $S$ that decides $A_{TM}:
S = \text{on input } \langle N, \omega \rangle$

1. Construct TM $M_2$ on input $\langle x \rangle$:
   1. If $x \neq \varepsilon$, reject.
   2. If $x = \varepsilon$, run $N$ on $\omega$ and accept if $N$ accepts.
2. Run $H$ on $\langle M_2 \rangle$. 

$M_2$ halts on $\varepsilon$\n$\Downarrow$\n$N$ accepts $\omega$
Claim: $HALT - \varepsilon_{TM} = \{\langle M \rangle: M \text{ is a TM and halts on empty input (i.e. } \varepsilon)\}$ is undecidable.

Proof: Suppose $HALT - \varepsilon_{TM}$ is decidable and let TM $H$ be its decider.

Build a TM $S$ that decides $A_{TM}$:

$S = \text{ on input } \langle N, \omega \rangle$

1. Construct TM $M_2$ on input $\langle x \rangle$:
   1. If $x \neq \varepsilon$, reject.
   2. If $x = \varepsilon$, run $N$ on $\omega$ and accept if $N$ accepts.

2. Run $H$ on $\langle M_2 \rangle$.

$M_2$ halts on $\varepsilon$
$\downarrow$
$N$ accepts $\omega$

rejects
Claim: $\text{HALT} - \varepsilon_{TM} = \{ \langle M \rangle : M \text{ is a TM and halts on empty input (i.e. } \varepsilon\} \text{ is undecidable.}

Proof: Suppose $\text{HALT} - \varepsilon_{TM}$ is decidable and let TM $H$ be its decider.

Build a TM $S$ that decides $A_{TM}$:

$S = \text{on input } \langle N, \omega \rangle$

1. Construct TM $M_2$ on input $\langle x \rangle$:
   1. If $x \neq \varepsilon$, reject.
   2. If $x = \varepsilon$, run $N$ on $\omega$ and accept if $N$ accepts.
2. Run $H$ on $\langle M_2 \rangle$.

$M_2$ halts on $\varepsilon$ ⇐ $N$ accepts $\omega$  \hspace{1cm} \text{rejects}$

Doesn’t work! $N$ not accepting is not the same as $N$ rejecting!
Claim: $HALT - \varepsilon_{TM} = \{\langle M \rangle: M$ is a TM and halts on empty input (i.e. $\varepsilon)\}$ is undecidable.

Proof: Suppose $HALT - \varepsilon_{TM}$ is decidable and let TM $H$ be its decider.

Build a TM $S$ that decides $HALT_{TM}$:

$S = \text{on input } \langle N, \omega \rangle$

1. Construct TM $M_2$ on input $\langle x \rangle$:

   1. Run $N$ on $\omega$ and accept if $N$ does.

2. Run $H$ on $\langle M_2 \rangle$. 

$M_2$ halts on $\varepsilon$

$\iff$

$N$ halts on $\omega$
Claim: $HALT - \varepsilon_{TM} = \{\langle M \rangle: M \text{ is a TM and halts on empty input (i.e. } \varepsilon)\}$ is undecidable.

Proof: Suppose $HALT - \varepsilon_{TM}$ is decidable and let TM $H$ be its decider.

Build a TM $S$ that decides $HALT_{TM}$:

$S = \text{on input } \langle N, \omega \rangle$

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

Doesn’t work! $M_2$ does not (always) halt when $N$ halts on $\omega$!
Claim: $\text{HALT} - \varepsilon_{TM} = \{\langle M \rangle : M \text{ is a TM and halts on empty input (i.e. } \varepsilon)\}$ is undecidable.

Proof: Suppose $\text{HALT} - \varepsilon_{TM}$ is decidable and let TM $H$ be its decider.

Build a TM $S$ that decides $\text{HALT}_{TM}$:

$S = \text{on input } \langle N, \omega \rangle$

1. Construct TM $M_2$ on input $\langle x \rangle$:
   1. Run $N$ on $\omega$.
   2. Accept if $N$ accepts and reject if $N$ rejects.

2. Run $H$ on $\langle M_2 \rangle$.

3. If $H$ accepts, accept. If $H$ rejects, reject.
Quiz 2 Logistics

1. During class on Thursday 10/27.

2. You can bring your book and any notes you would like, but no electronic devices.

3. You may assume anything proven in class or on homeworks.

4. 4 questions:
   1) Show a language is decidable.
   2) Show a language is not decidable x2.
   3) Conceptual question.
Prove that $ALMOST\_ALL_{DFA} = \{(A, \omega): A \text{ is a DFA and } L(A) = \Sigma^* \setminus \{\omega\}\}$ is decidable. Remember to show that your algorithm meets the requirements of a decider.

You are allowed to use any deciders we learned in class or on homework!
Prove that $ALMOST\_ALL_{DFA} = \{\langle A, \omega \rangle: A \text{ is a DFA and } L(A) = \Sigma^* \setminus \{\omega\}\}$ is decidable. Remember to show that your algorithm meets the requirements of a decider.

Proof: Construct the following Turing machine to decide $ALMOST\_ALL_{DFA}$:

$M = \text{on input } \langle A, \omega \rangle$

1.
Prove that $ALMOST\_ALL_{DFA} = \{\langle A, \omega \rangle: A \text{ is a DFA and } L(A) = \Sigma^* \setminus \{\omega\}\}$ is decidable. Remember to show that your algorithm meets the requirements of a decider.

Proof: Construct the following Turing machine to decide $ALMOST\_ALL_{DFA}$:

$M = \text{ on input } \langle A, \omega \rangle$

1. Run $A$ on $\omega$ and reject if $A$ accepts.
Prove that $ALMOST\_ALL_{DFA} = \{\langle A, \omega \rangle: A \text{ is a DFA and } L(A) = \Sigma^* \setminus \{\omega\}\}$ is decidable. Remember to show that your algorithm meets the requirements of a decider.

Proof: Construct the following Turing machine to decide $ALMOST\_ALL_{DFA}$: 

$M = \text{on input } \langle A, \omega \rangle$

1. Run $A$ on $\omega$ and reject if $A$ accepts.
2. Construct DFA $B$ that only accepts $\omega$. 
Prove that $ALMOST\_ALL_{DFA} = \{(A, \omega): A \text{ is a DFA and } L(A) = \Sigma^* \setminus \{\omega\} \}$ is decidable. Remember to show that your algorithm meets the requirements of a decider.

Proof: Construct the following Turing machine to decide $ALMOST\_ALL_{DFA}$: 

$M = \text{on input } \langle A, \omega \rangle$

1. Run $A$ on $\omega$ and reject if $A$ accepts.
2. Construct DFA $B$ that only accepts $\omega$.
3. Construct DFA $C$ so that $L(C) = L(A) \cup L(B)$.
Prove that $ALMOST\_ALL_{DFA} = \{\langle A, \omega \rangle: A \text{ is a DFA and } L(A) = \Sigma^* \setminus \{\omega\}\}$ is decidable. Remember to show that your algorithm meets the requirements of a decider.

Proof: Construct the following Turing machine to decide $ALMOST\_ALL_{DFA}$:

$M = \text{ on input } \langle A, \omega \rangle$

1. Run $A$ on $\omega$ and reject if $A$ accepts.
2. Construct DFA $B$ that only accepts $\omega$.
3. Construct DFA $C$ so that $L(C) = L(A) \cup L(B)$.
4. Run the decider for $ALL_{DFA}$ on $C$. 
Prove that \( ALMOST\_ALL_{DFA} = \{ \langle A, \omega \rangle : A \text{ is a DFA and } L(A) = \Sigma^* \setminus \{ \omega \} \} \) is decidable. Remember to show that your algorithm meets the requirements of a decider.

Proof: Construct the following Turing machine to decide \( ALMOST\_ALL_{DFA} \):

\[ M = \text{on input } \langle A, \omega \rangle \]

1. Run \( A \) on \( \omega \) and reject if \( A \) accepts.
2. Construct DFA \( B \) that only accepts \( \omega \).
3. Construct DFA \( C \) so that \( L(C) = L(A) \cup L(B) \).
4. Run the decider for \( ALL_{DFA} \) on \( C \).
5. If the decider accepts, ???.

Prove that $ALMOST\_ALL_{DFA} = \{\langle A, \omega \rangle: A \text{ is a DFA and } L(A) = \Sigma^* \setminus \{\omega\}\}$ is decidable. Remember to show that your algorithm meets the requirements of a decider.

Proof: Construct the following Turing machine to decide $ALMOST\_ALL_{DFA}$:

$M = \text{on input } \langle A, \omega \rangle$

1. Run $A$ on $\omega$ and reject if $A$ accepts.
2. Construct DFA $B$ that only accepts $\omega$.
3. Construct DFA $C$ so that $L(C) = L(A) \cup L(B)$.
4. Run the decider for $ALL_{DFA}$ on $C$.
5. If the decider accepts, accept. If it rejects, reject.
Prove that $ALMOST\_ALL_{DFA} = \{\langle A, \omega \rangle : A \text{ is a DFA and } L(A) = \Sigma^* \backslash \{\omega\}\}$ is decidable. Remember to show that your algorithm meets the requirements of a decider.

Proof: Construct the following Turing machine to decide $ALMOST\_ALL_{DFA}$:

$M = \text{on input } \langle A, \omega \rangle$

1. Run $A$ on $\omega$ and \textbf{reject} if $A$ accepts.
2. Construct DFA $B$ that only accepts $\omega$.
3. Construct DFA $C$ so that $L(C) = L(A) \cup L(B)$.
4. Run the decider for $ALL_{DFA}$ on $C$.
5. If the decider accepts, \textbf{accept}. If it rejects, \textbf{reject}.

Every DFA is guaranteed to halt on all input so step 1 will halt. Constructing $B$ and $C$ will occur in finite time, and running a decider is guaranteed to halt. Thus, $M$ is guaranteed to halt and is a decider.
Prove that $ANY_{TM} = \{\langle M \rangle: M \text{ is a TM and } L(M) \neq \emptyset \}$ is undecidable.
Prove that $ANY_{TM} = \{\langle M \rangle : M \text{ is a TM and } L(M) \neq \emptyset \}$ is undecidable.

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

Build a TM $S$ that decides $A_{TM}$:

$S = \text{ on input } \langle N, \omega \rangle$

1. Construct TM $M_2$ on input $\langle x \rangle$:
   - ???

2. Run $H$ on $\langle M_2 \rangle$.
3. If $H$ accepts, ??? If $H$ rejects, ???.

Justification...
Prove that $ANY_{TM} = \{ \langle M \rangle : M \text{ is a TM and } L(M) \neq \emptyset \}$ is undecidable.

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

Build a TM $S$ that decides $A_{TM}$:

$S = \text{on input } \langle N, \omega \rangle$:

1. Construct TM $M_2$ on input $\langle x \rangle$:
   - $\text{ ???}$

2. Run $H$ on $\langle M_2 \rangle$.

3. If $H$ accepts, $\text{ ???}$. If $H$ rejects, $\text{ ???}$.

Justification...

$L(M_2) \neq \emptyset \iff N \text{ accepts } \omega$
Prove that \( \text{ANY}_{TM} = \{\langle M \rangle : M \text{ is a TM and } L(M) \neq \emptyset \} \) is undecidable.

Proof: Suppose \( \text{ANY}_{TM} \) is decidable and let TM \( H \) be its decider.

Build a TM \( S \) that decides \( A_{TM} : \)
\[
S = \text{on input } \langle N, \omega \rangle \\
1. \text{Construct TM } M_2 \text{ on input } \langle x \rangle : \\
   1. \text{Run } N \text{ on } \omega \text{ and accept if } N \text{ does.} \\
   2. \text{Run } H \text{ on } \langle M_2 \rangle. \\
   3. \text{If } H \text{ accepts, } L(M_2) \neq \emptyset \text{. If } H \text{ rejects, } \emptyset. \\
\]

Justification...
Prove that \( \text{ANY}_{TM} = \{ \langle M \rangle : M \text{ is a TM and } L(M) \neq \emptyset \} \) is undecidable.

Proof: Suppose \( \text{ANY}_{TM} \) is decidable and let TM \( H \) be its decider.

Build a TM \( S \) that decides \( A_{TM} \):

\[
S = \text{on input } \langle N, \omega \rangle
\]

1. Construct TM \( M_2 \) on input \( \langle x \rangle \):
   1. Run \( N \) on \( \omega \) and accept if \( N \) does.
   2. Run \( H \) on \( \langle M_2 \rangle \).
   3. If \( H \) accepts, accept. If \( H \) rejects, ???.

Justification...
Prove that \( \text{ANY}_{TM} = \{\langle M \rangle : M \text{ is a TM and } L(M) \neq \emptyset \} \) is undecidable.

Proof: Suppose \( \text{ANY}_{TM} \) is decidable and let TM \( H \) be its decider.

Build a TM \( S \) that decides \( A_{TM} \):

\[ S = \text{on input } \langle N, \omega \rangle \]

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

Justification...
Prove that $ANY_{TM} = \{ \langle M \rangle : M \text{ is a TM and } L(M) \neq \emptyset \}$ is undecidable.

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

Build a TM $S$ that decides $A_{TM}$:

$S = \text{on input } \langle N, \omega \rangle$

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

If $N$ accepts $\omega$, $L(M_2) = \Sigma^*$ and $S$ accepts. If $N$ does not accept $\omega$, $L(M_2) = \emptyset$ and $S$ rejects. Thus, $S$ is a decider for $A_{TM}$, which is a contradiction, so $ANY_{TM}$ is undecidable.