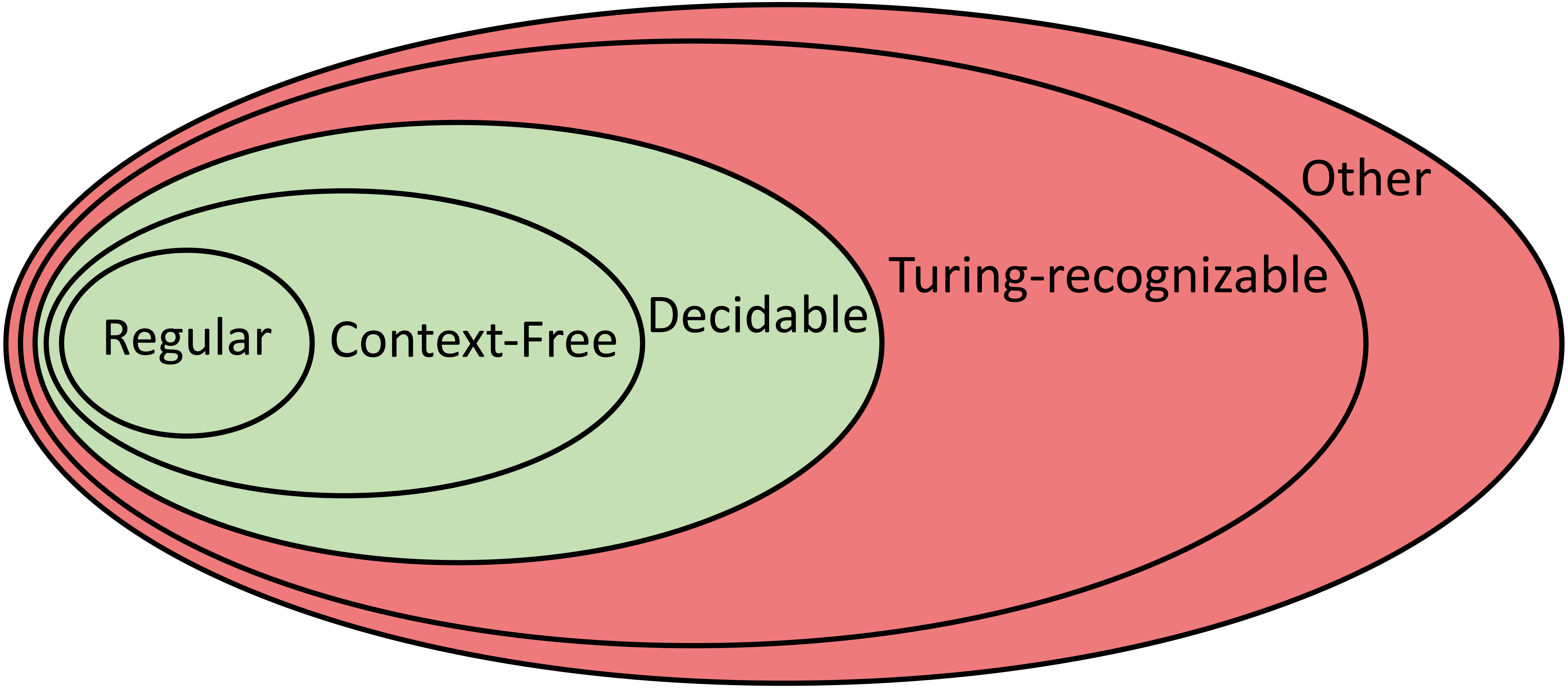
Decidability

CSCI 338

Decidable

A language L is <u>decidable</u> if there is a TM (a decider) that accepts every string in the language and rejects everything else. (i.e. halts on all input)

Computability Hierarchy



EQ_{DFA}

Claim: $EQ_{DFA} = \{\langle A, B\rangle : A \text{ and } B \text{ are DFAs and } L(A) = L(B)\}$ is decidable.

Proof:

?

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

EQ_{DFA}

Claim: $EQ_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(B)\}$ is decidable.

Proof:

What if we tried to use E_{DFA} somehow?

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

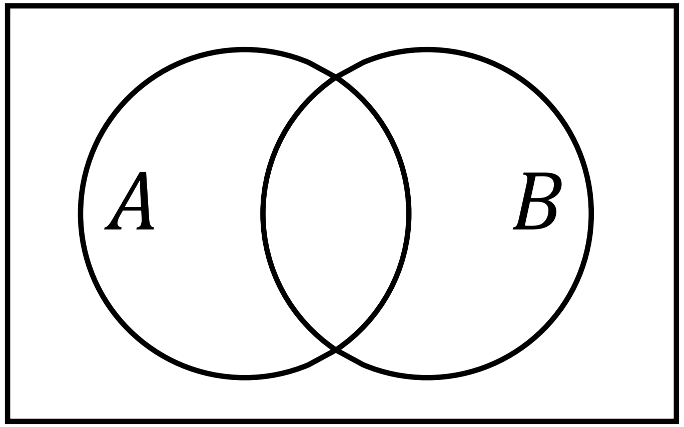
EQ_{DFA}

Claim: $EQ_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(B)\}$ is decidable.

Proof:

What if we tried to use E_{DFA} somehow?

If $L(A) = L(B)$, what would be empty?



A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

EQ_{DFA}

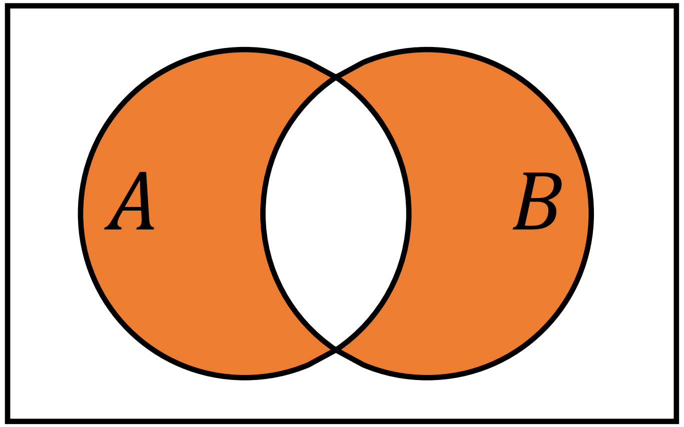
Claim: $EQ_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(B)\}$ is decidable.

Proof:

What if we tried to use E_{DFA} somehow?

If $L(A) = L(B)$, what would be empty?

The part of $L(A)$ not in $L(B)$ and the part of $L(B)$ not in $L(A)$.



A language L is decidable if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

EQ_{DFA}

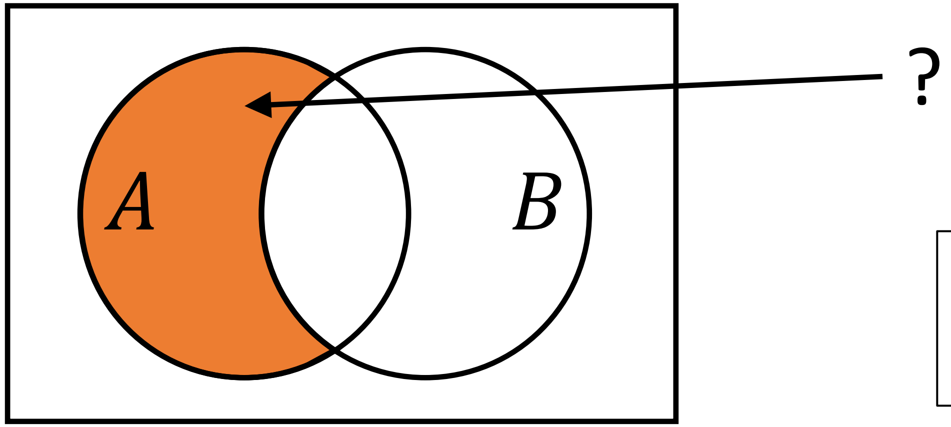
Claim: $EQ_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(B)\}$ is decidable.

Proof:

What if we tried to use E_{DFA} somehow?

If $L(A) = L(B)$, what would be empty?

The part of $L(A)$ not in $L(B)$ and the part of $L(B)$ not in $L(A)$.



A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

EQ_{DFA}

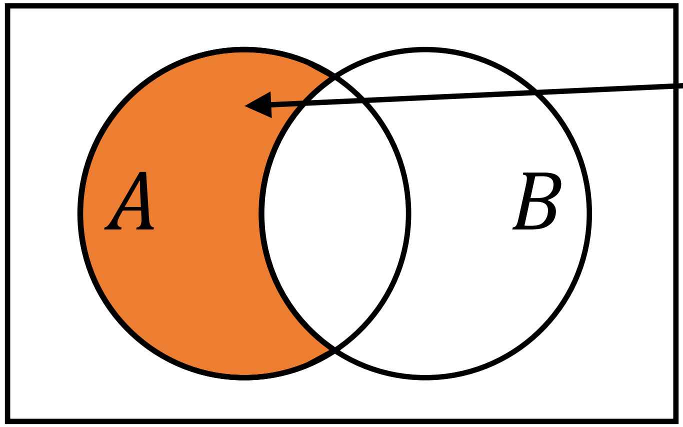
Claim: $EQ_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(B)\}$ is decidable.

Proof:

What if we tried to use E_{DFA} somehow?

If $L(A) = L(B)$, what would be empty?

The part of $L(A)$ not in $L(B)$ and the part of $L(B)$ not in $L(A)$.



$A \cap \bar{B}$

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

EQ_{DFA}

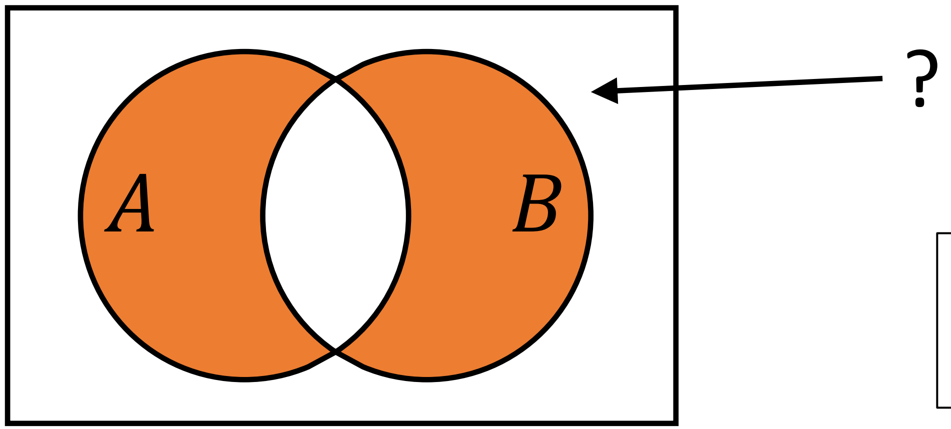
Claim: $EQ_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(B)\}$ is decidable.

Proof:

What if we tried to use E_{DFA} somehow?

If $L(A) = L(B)$, what would be empty?

The part of $L(A)$ not in $L(B)$ and the part of $L(B)$ not in $L(A)$.



A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

EQ_{DFA}

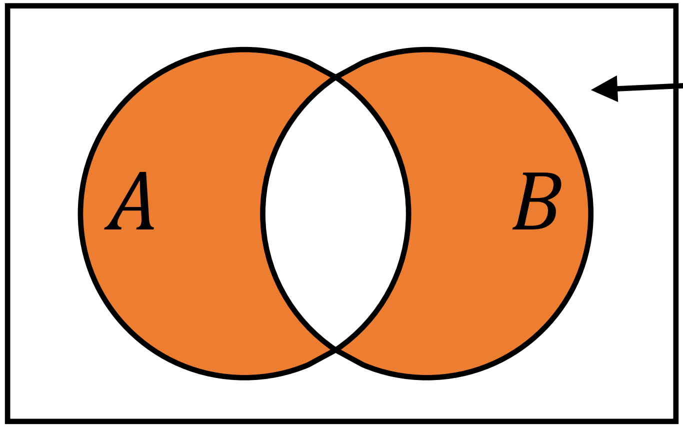
Claim: $EQ_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(B)\}$ is decidable.

Proof:

What if we tried to use E_{DFA} somehow?

If $L(A) = L(B)$, what would be empty?

The part of $L(A)$ not in $L(B)$ and the part of $L(B)$ not in $L(A)$.



$(A \cap \bar{B}) \cup (\bar{A} \cap B)$

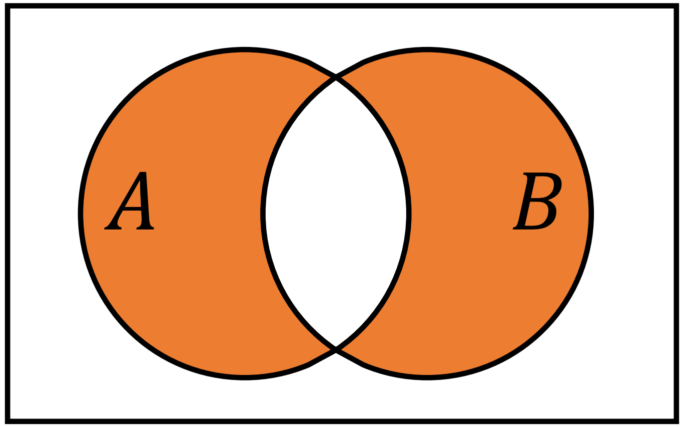
A language L is decidable if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

EQ_{DFA}

Claim: $EQ_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(B)\}$ is decidable.

Proof:

M$_4$ = on input $\langle A, B\rangle$



A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

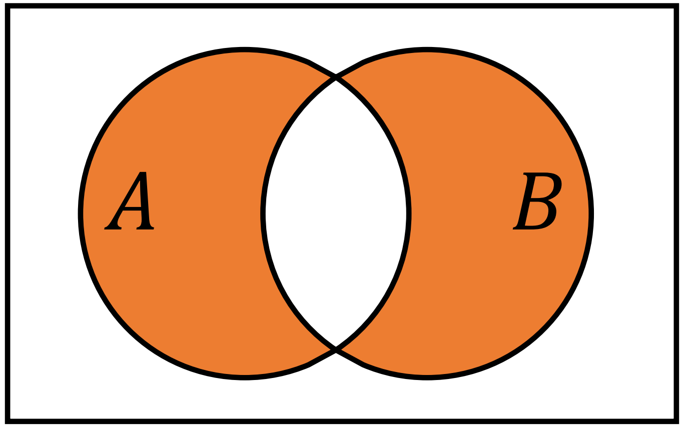
EQ_{DFA}

Claim: $EQ_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(B)\}$ is decidable.

Proof:

M$_4$ = on input $\langle A, B\rangle$

1. Construct DFA C as $(A \cap \bar{B}) \cup (\bar{A} \cap B)$.



A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

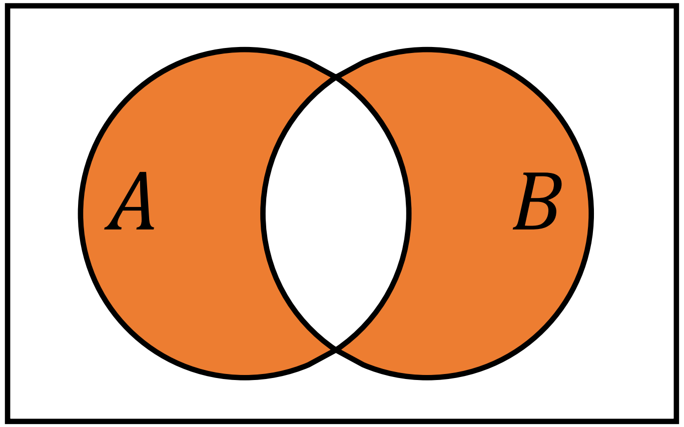
EQ_{DFA}

Claim: $EQ_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(B)\}$ is decidable.

Proof:

M$_4$ = on input $\langle A, B\rangle$

1. Construct DFA C as $\overbrace{(A \cap \bar{B}) \cup (\bar{A} \cap B)}^{\textbf{Details???}}$.



A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

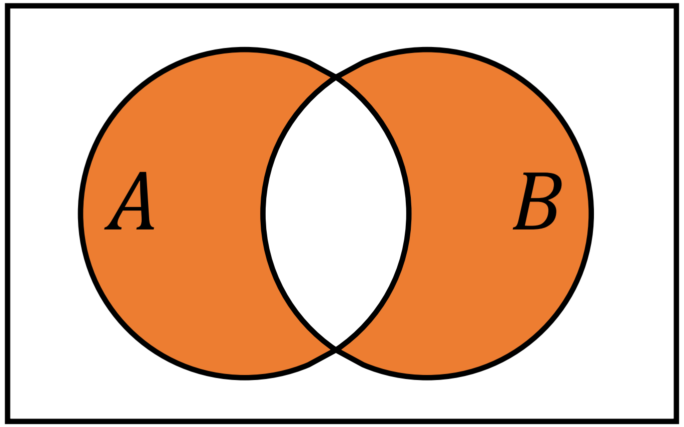
EQ_{DFA}

Claim: $EQ_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(B)\}$ is decidable.

Proof:

M$_4$ = on input $\langle A, B\rangle$

1. Construct DFA C as $(A \cap \bar{B}) \cup (\bar{A} \cap B)$.
2. Run E_{DFA} Decider on $\langle C\rangle$.



A language L is decidable if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

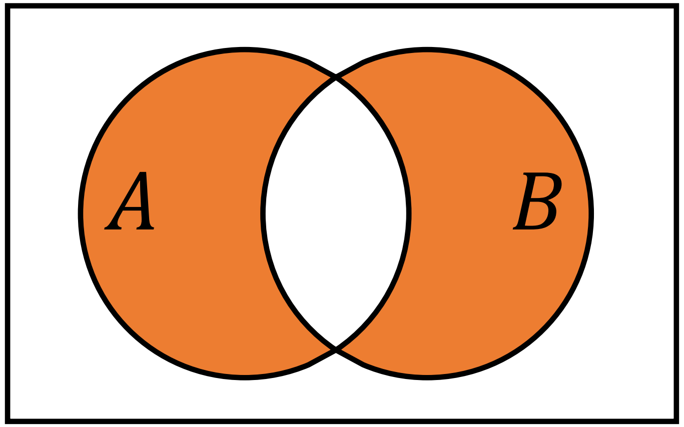
EQ_{DFA}

Claim: $EQ_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(B)\}$ is decidable.

Proof:

M_4 = on input $\langle A, B\rangle$

1. Construct DFA C as $(A \cap \bar{B}) \cup (\bar{A} \cap B)$.
2. Run E_{DFA} Decider on $\langle C\rangle$.
3. Accept/Reject?



A language L is decidable if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

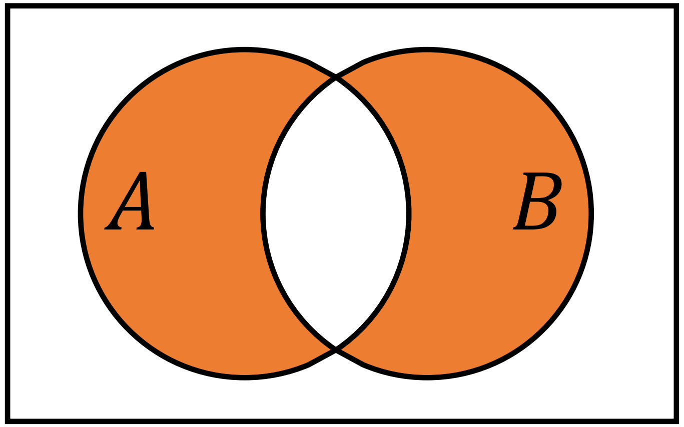
EQ_{DFA}

Claim: $EQ_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(B)\}$ is decidable.

Proof:

M_4 = on input $\langle A, B\rangle$

1. Construct DFA C as $(A \cap \bar{B}) \cup (\bar{A} \cap B)$.
2. Run E_{DFA} Decider on $\langle C\rangle$.
3. If Decider accepts, <u>accept</u>. If Decider rejects, <u>reject</u>.



M_4 is a decider since constructing C halts and the E_{DFA} Decider is a decider.

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A \text{ is a DFA and } |L(A)| = \infty\}$ is decidable.

Proof:

?

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A$ is a DFA and $|L(A)| = \infty\}$ is decidable.

Proof:

$$|L(A)| = \infty \Leftrightarrow \exists \text{ loops in } A.$$

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A \text{ is a DFA and } |L(A)| = \infty\}$ is decidable.

Proof:

$|L(A)| = \infty \Leftrightarrow \exists$ loops in A.
loops in $A \Leftrightarrow A$ accepts strings $\geq$???

A language L is decidable if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A \text{ is a DFA and } |L(A)| = \infty\}$ is decidable.

Proof:

$|L(A)| = \infty \Leftrightarrow \exists$ loops in A.
loops in $A \Leftrightarrow A$ accepts strings $\geq$ # states in A.

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A$ is a DFA and $|L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A \rangle$

1. Let p be number of states in A.

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A \text{ is a DFA and } |L(A)| = \infty\}$ is decidable.

Proof:

M_5 = on input $\langle A \rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

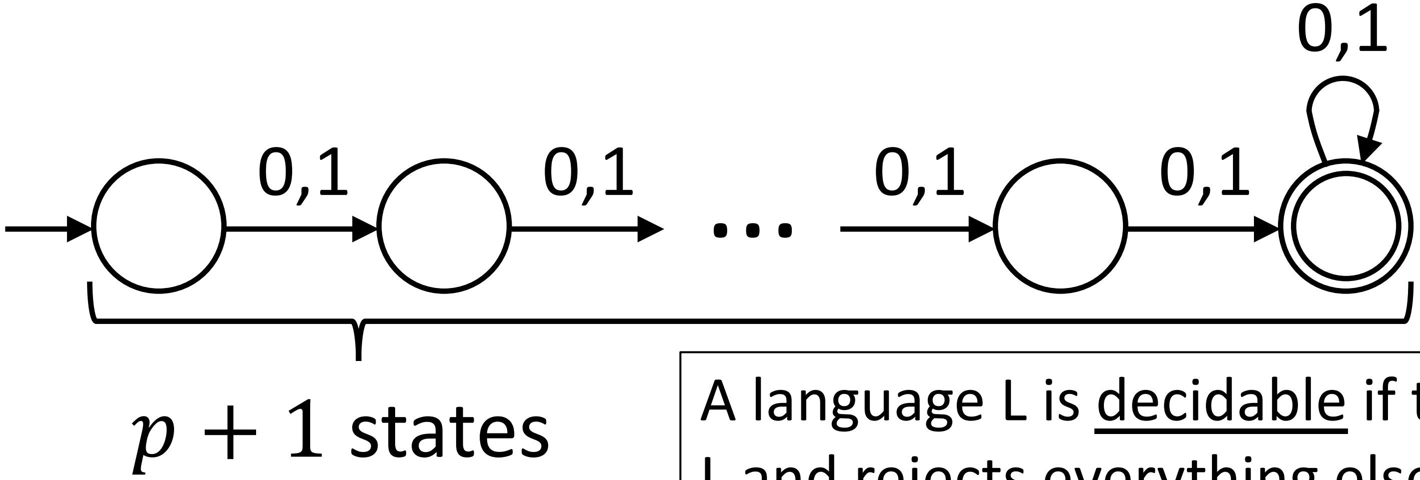
$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A\rangle : A \text{ is a DFA and } |L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A\rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.



A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A\rangle : A \text{ is a DFA and } |L(A)| = \infty\}$ is decidable.

Proof:

M_5 = on input $\langle A\rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A$ is a DFA and $|L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A \rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.
4. ?

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A\rangle : A$ is a DFA and $|L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A\rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.
4. Run E_{DFA} Decider on $\langle M\rangle$.

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A$ is a DFA and $|L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A \rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.
4. Run E_{DFA} Decider on $\langle M \rangle$.
5. If Decider accepts, <u>?</u>

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A$ is a DFA and $|L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A \rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.
4. Run E_{DFA} Decider on $\langle M \rangle$. **If the E_{DFA} Decider accepts, ???**
5. If Decider accepts, ?

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A \text{ is a DFA and } |L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A \rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.
4. Run E_{DFA} Decider on $\langle M \rangle$. **If the E_{DFA} Decider accepts, $L(M)$ is empty**
5. If Decider accepts, ?

$$E_{DFA} = \{\langle A \rangle : A \text{ is a DFA and } L(A) = \emptyset\}$$

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A$ is a DFA and $|L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A \rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.
4. Run E_{DFA} Decider on $\langle M \rangle$.
5. If Decider accepts, ?

If the E_{DFA} Decider accepts, $L(M)$ is empty

$\Rightarrow$ No string is in both $L(A)$ and $L(D)$

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A$ is a DFA and $|L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A \rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.
4. Run E_{DFA} Decider on $\langle M \rangle$.
5. If Decider accepts, ?

If the E_{DFA} Decider accepts, $L(M)$ is empty

$\Rightarrow$ No string is in both $L(A)$ and $L(D)$

$\Rightarrow$ No strings in $L(A)$ are $\geq p$ characters

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A\rangle : A$ is a DFA and $|L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A\rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.
4. Run E_{DFA} Decider on $\langle M\rangle$.
5. If Decider accepts, ?

If the E_{DFA} Decider accepts, $L(M)$ is empty

$\Rightarrow$ No string is in both $L(A)$ and $L(D)$

$\Rightarrow$ No strings in $L(A)$ are $\geq p$ characters

$\Rightarrow$ All strings in $L(A)$ are $< p$ characters

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A\rangle : A$ is a DFA and $|L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A\rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.
4. Run E_{DFA} Decider on $\langle M\rangle$.
5. If Decider accepts, ?

If the E_{DFA} Decider accepts, $L(M)$ is empty
$\Rightarrow$ No string is in both $L(A)$ and $L(D)$
$\Rightarrow$ No strings in $L(A)$ are $\geq p$ characters
$\Rightarrow$ All strings in $L(A)$ are $< p$ characters
$\Rightarrow$ $L(A)$ must be ???

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A$ is a DFA and $|L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A \rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.
4. Run E_{DFA} Decider on $\langle M \rangle$.
5. If Decider accepts, <u>?</u>

If the E_{DFA} Decider accepts, $L(M)$ is empty
$\Rightarrow$ No string is in both $L(A)$ and $L(D)$
$\Rightarrow$ No strings in $L(A)$ are $\geq p$ characters
$\Rightarrow$ All strings in $L(A)$ are $< p$ characters
$\Rightarrow$ $L(A)$ must be finite in size

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A$ is a DFA and $|L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A \rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.
4. Run E_{DFA} Decider on $\langle M \rangle$.
5. If Decider accepts, ?

If the E_{DFA} Decider accepts, $L(M)$ is empty
$\Rightarrow$ No string is in both $L(A)$ and $L(D)$
$\Rightarrow$ No strings in $L(A)$ are $\geq p$ characters
$\Rightarrow$ All strings in $L(A)$ are $< p$ characters
$\Rightarrow$ $L(A)$ must be finite in size
$\Rightarrow$???

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A$ is a DFA and $|L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A \rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.
4. Run E_{DFA} Decider on $\langle M \rangle$.
5. If Decider accepts, reject.

If the E_{DFA} Decider accepts, $L(M)$ is empty
$\Rightarrow$ No string is in both $L(A)$ and $L(D)$
$\Rightarrow$ No strings in $L(A)$ are $\geq p$ characters
$\Rightarrow$ All strings in $L(A)$ are $< p$ characters
$\Rightarrow$ $L(A)$ must be finite in size
$\Rightarrow$ Reject!!!

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A \text{ is a DFA and } |L(A)| = \infty\}$ is decidable.

Proof:

M$_5$ = on input $\langle A \rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.
4. Run E_{DFA} Decider on $\langle M \rangle$.
5. If Decider accepts, <u>reject</u>.

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$INFINITE_{DFA}$

Claim: $INFINITE_{DFA} = \{\langle A \rangle : A \text{ is a DFA and } |L(A)| = \infty\}$ is decidable.

Proof:

M_5 = on input $\langle A \rangle$

1. Let p be number of states in A.
2. Construct DFA D that accepts all strings of length $\geq p$.
3. Construct DFA M where $L(M) = L(A) \cap L(D)$.
4. Run E_{DFA} Decider on $\langle M \rangle$.
5. If Decider accepts, <u>reject</u>. If Decider rejects, <u>accept</u>.

M_5 is a decider since D and M are finite and the E_{DFA} Decider is a decider.

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$COMPLEMENTS_{DFA}$

Claim: $COMPLEMENTS_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(\bar{B})\}$ is decidable.

Proof:

?

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$COMPLEMENTS_{DFA}$

Claim: $COMPLEMENTS_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(\bar{B})\}$ is decidable.

Proof:

M$_6$ = on input $\langle A, B\rangle$

1. Let $C = \bar{B}$.

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$COMPLEMENTS_{DFA}$

Claim: $COMPLEMENTS_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(\bar{B})\}$ is decidable.

Proof:

M$_6$ = on input $\langle A, B\rangle$

1. Let $C = \bar{B}$.
2. Run EQ_{DFA} Decider on $\langle A, C\rangle$.

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$COMPLEMENTS_{DFA}$

Claim: $COMPLEMENTS_{DFA} = \{\langle A, B\rangle \colon A$ and B are DFAs and $L(A) = L(\bar{B})\}$ is decidable.

Proof:

M$_6$ = on input $\langle A, B\rangle$

1. Let $C = \bar{B}$.
2. Run EQ_{DFA} Decider on $\langle A, C\rangle$.
3. If Decider accepts, <u>???</u>.

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$COMPLEMENTS_{DFA}$

Claim: $COMPLEMENTS_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(\bar{B})\}$ is decidable.

Proof:

M$_6$ = on input $\langle A, B\rangle$

1. Let $C = \bar{B}$.
2. Run EQ_{DFA} Decider on $\langle A, C\rangle$.
3. If Decider accepts, <u>accept</u>. If Decider rejects, <u>reject</u>.

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

$COMPLEMENTS_{DFA}$

Claim: $COMPLEMENTS_{DFA} = \{\langle A, B\rangle : A$ and B are DFAs and $L(A) = L(\bar{B})\}$ is decidable.

Proof:

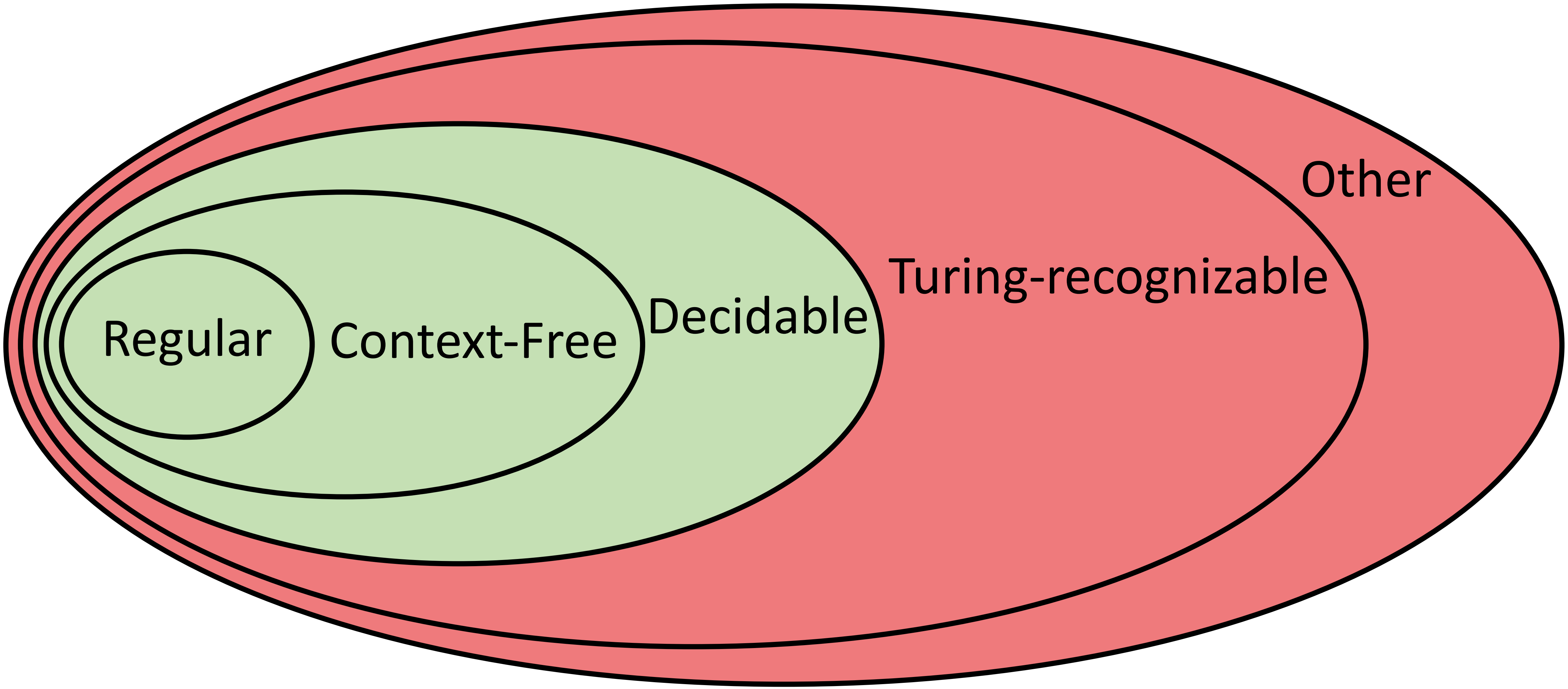
M$_6$ = on input $\langle A, B\rangle$

1. Let $C = \bar{B}$.
2. Run EQ_{DFA} Decider on $\langle A, C\rangle$.
3. If Decider accepts, <u>accept</u>. If Decider rejects, <u>reject</u>.

M$_6$ is a decider since constructing C halts and the EQ_{DFA} Decider is a decider.

A language L is <u>decidable</u> if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)

Computability Hierarchy



A_{TM}

Claim: $A_{TM} = \{\langle M, \omega \rangle : M$ is a TM and M accepts ω} is decidable.

Proof:

?

A language L is decidable if there is a TM that recognizes L and rejects everything else. (i.e. halts on all input)