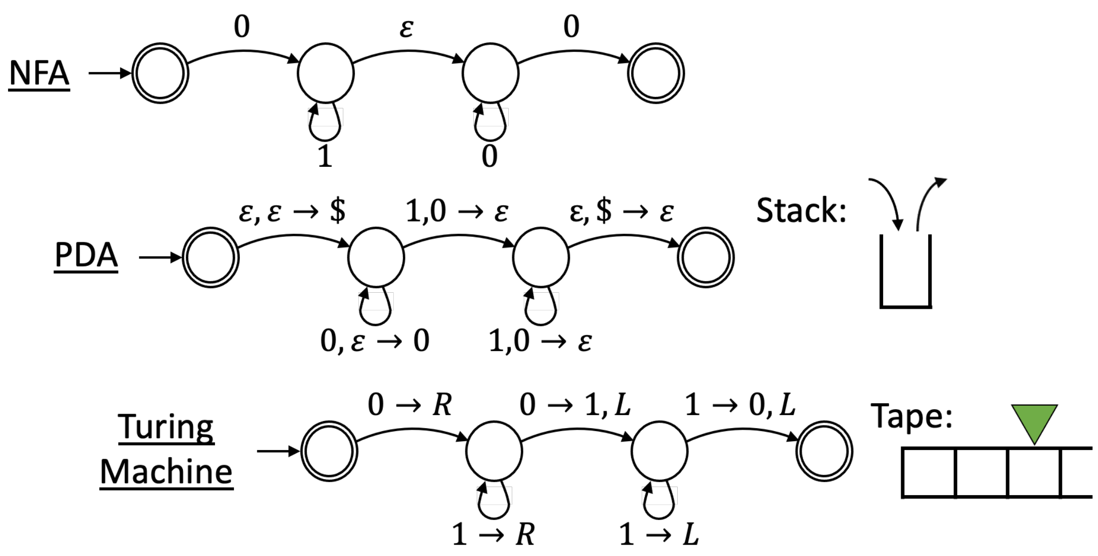
Decidability

CSCI 338

August → **December**

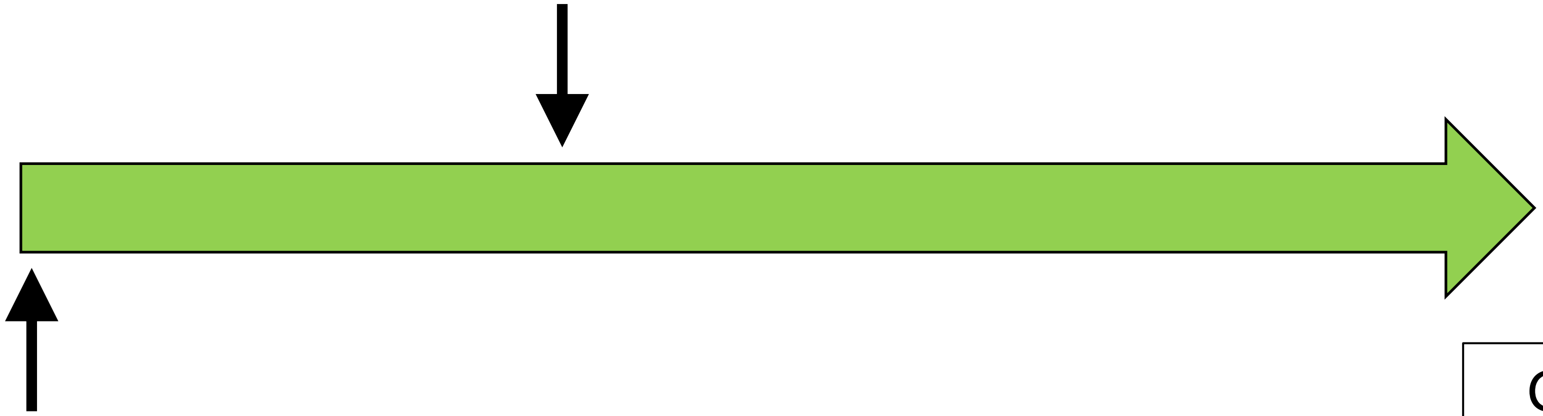
Computational Models

Goal: Understand and identify fundamental limitations of computers.



Computability:
What’s solvable
by computers.

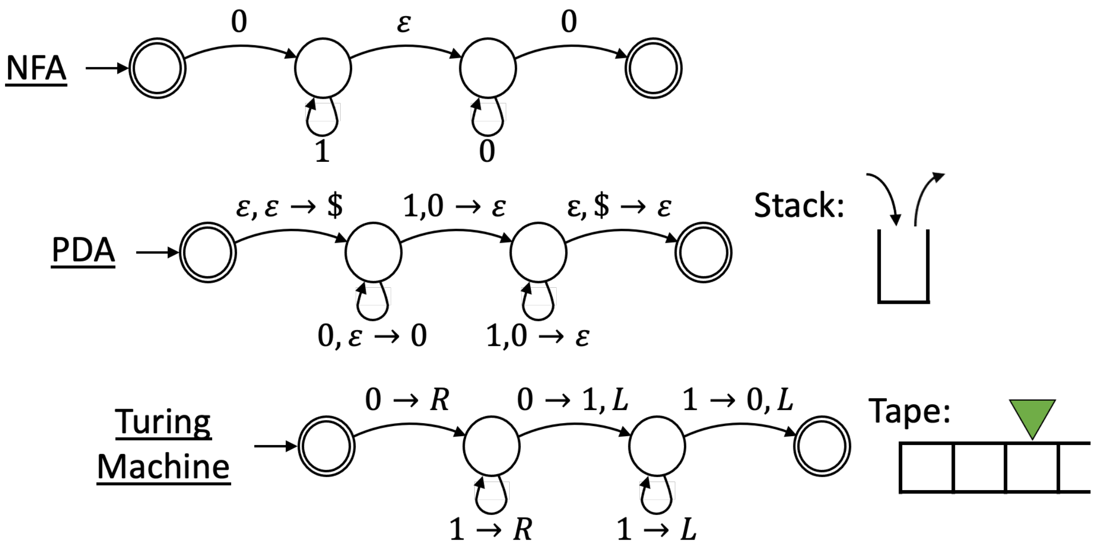
August



December

Computational
Models

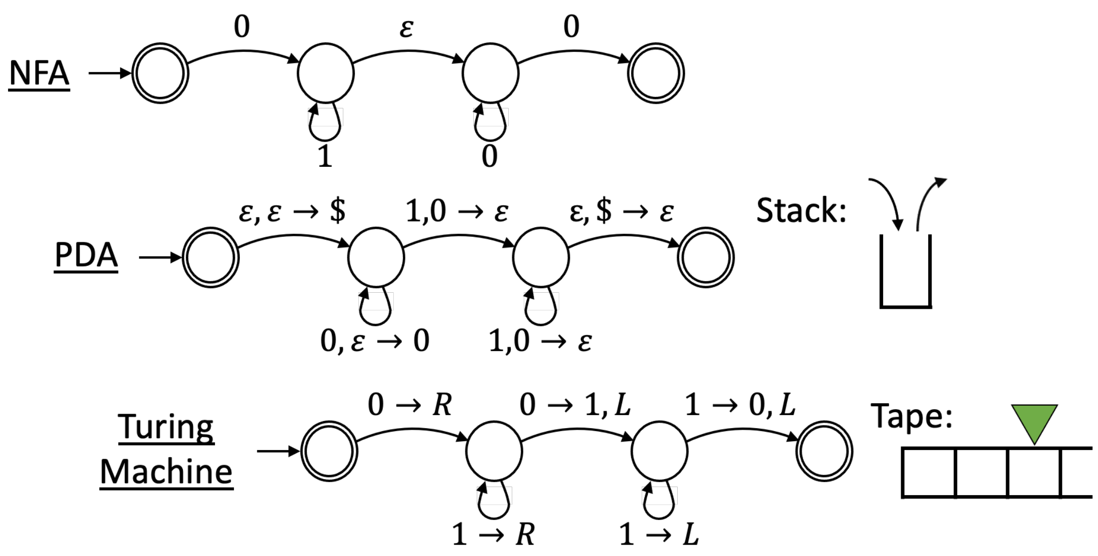
Goal: Understand and identify fundamental limitations of computers.



Computability: What's solvable by computers.

August → **December**

Computational Models



Complexity: What's <u>efficiently</u> solvable by computers.

<u>Goal:</u> Understand and identify fundamental limitations of computers.

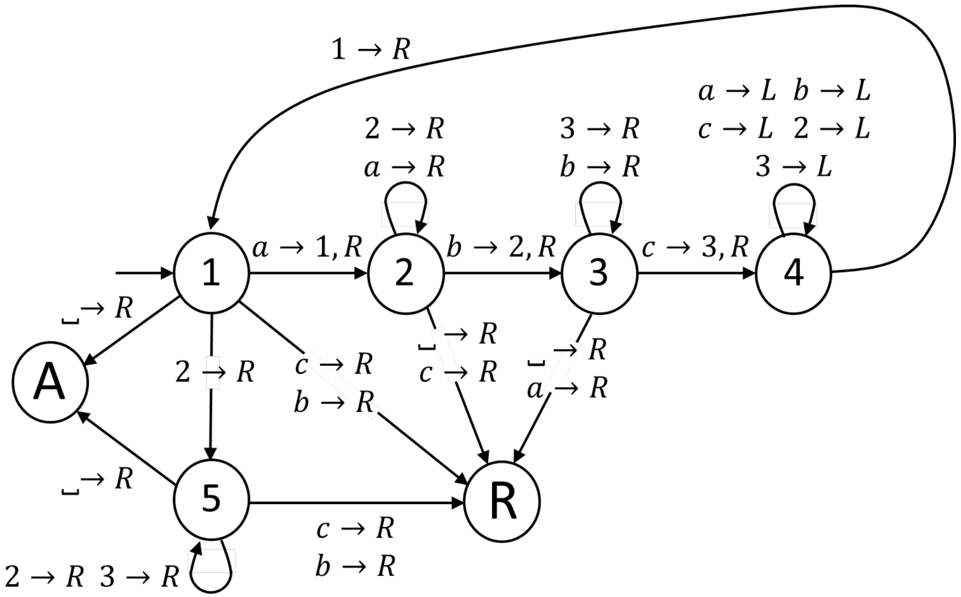
Church-Turing Thesis

Intuitive notion of algorithms. **=** Turing Machine algorithms.

TM M: on input ω

1. If $\omega = \varepsilon$, accept. Otherwise, change first a to a 1.
2. Move right to first b and change to a 2. Reject if c or ␣ found first.
3. Move right to first c and change to a 3. Reject if a or ␣ found first.
4. Move back to first a. If it exists, loop to step 1. If not, exit loop.
5. Move right to verify no b or c exist. If so, reject. If not, accept.

=



Definitions

A language is <u>Turing recognizable</u> if there is a TM that accepts every string in the language, and nothing not in the language.

Called 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)

Definitions

Language $L = \{w : |w| \text{ is even}\}$

A language is <u>Turing recogn</u>
accepts every string in the l
the language.

```
if (s.length() % 2 == 0)
{
    return true;
} else {
    return false;
}
```

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)

Definitions

Language $L = \{w: |w| \text{ is even}\}$

A language is Turing recognizable if there is a TM that accepts every string in the language, and nothing not in the language.

A language L is decidabl

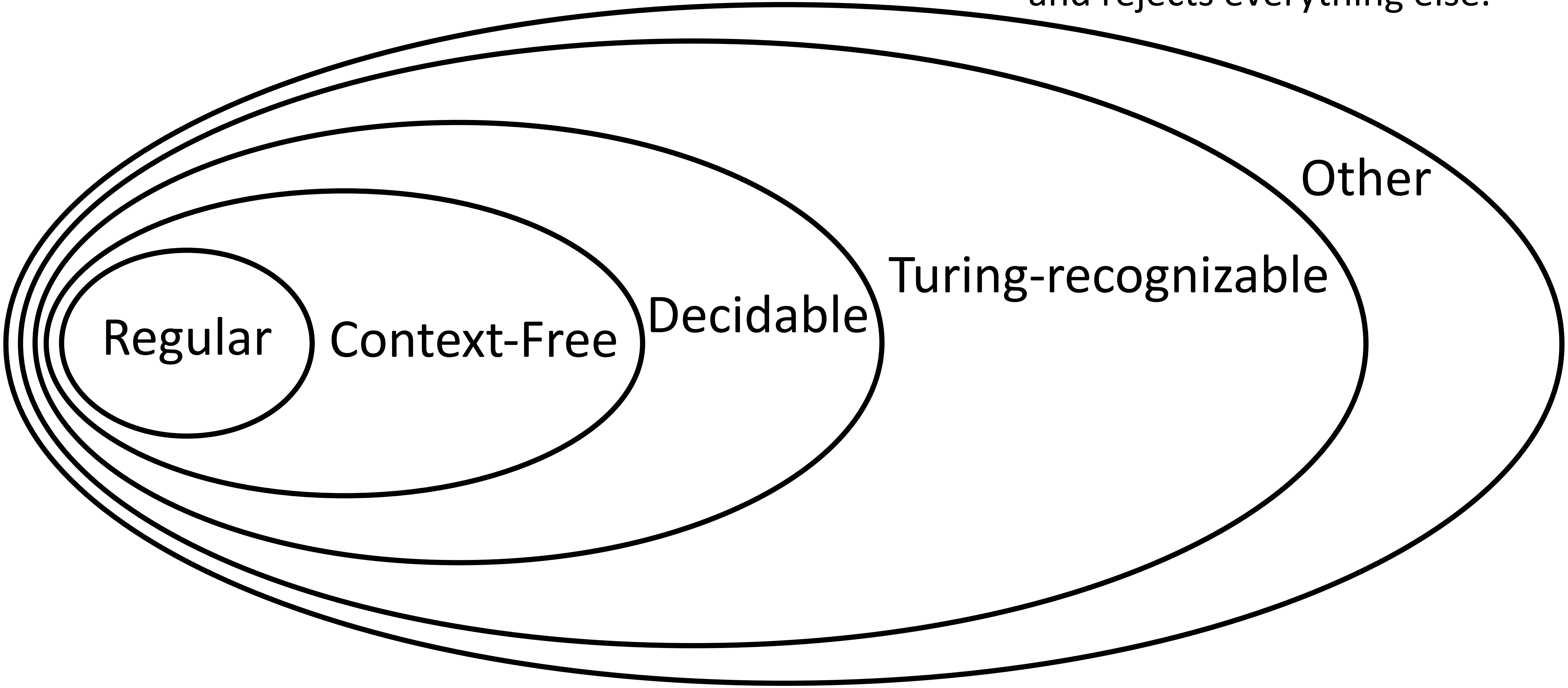
L and rejects everything

```
if (s.length() % 2 == 0) {
   return true;
} else {
   while (true) {
      contemplateMortality();
   }
}
```

Computability Hierarchy

Recognizable: ∃ TM that accepts everything in L, and nothing not.

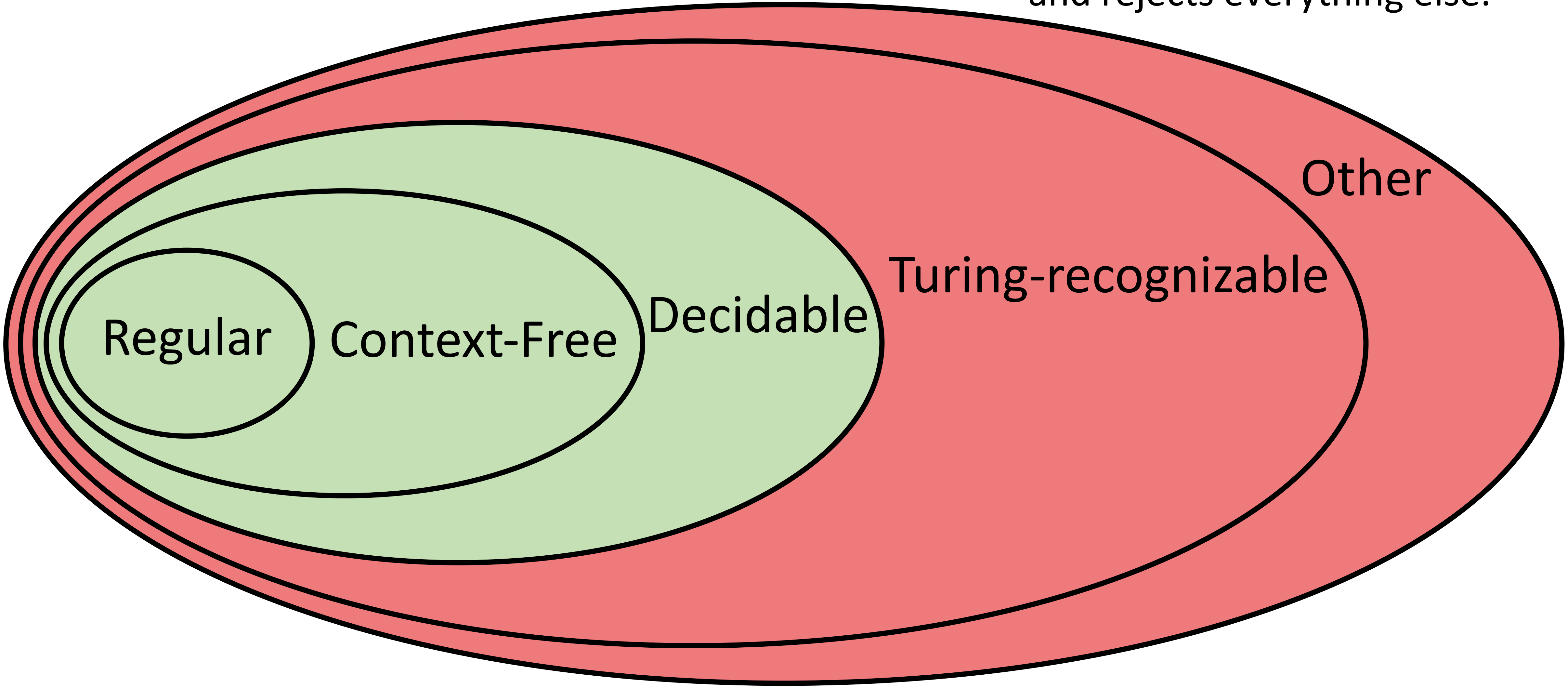
Decidable: ∃ TM that recognizes L and rejects everything else.



Computability Hierarchy

<u>Recognizable:</u> ∃ TM that accepts everything in L, and nothing not.

<u>Decidable:</u> ∃ TM that recognizes L and rejects everything else.



A_{DFA}

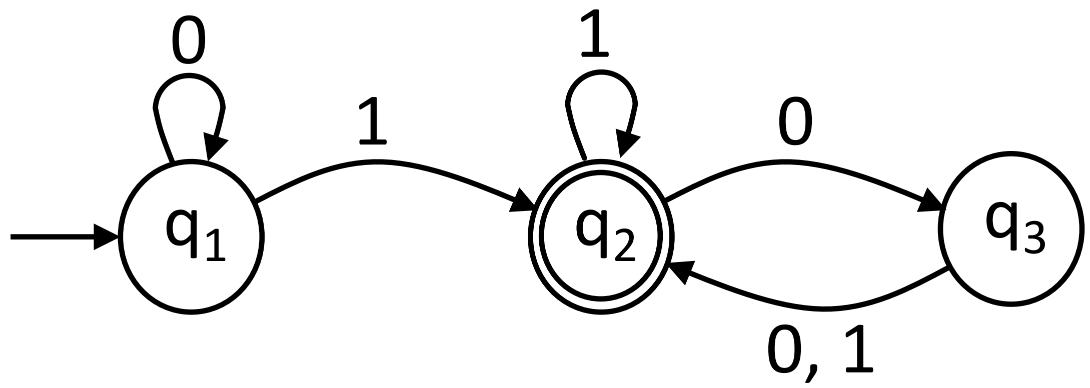
Claim: $A_{DFA} = \{\langle B, \omega \rangle : B$ is a DFA that accepts string $\omega\}$ is a decidable language.

A_{DFA}

Denotes string encoding of some object

Claim: $A_{DFA} = \{\langle B, \omega \rangle \mid B$ is a DFA that accepts string $\omega\}$ is a decidable language.

DFA Formal Definition



$Q = \{q_1, q_2, q_3\}$

$\Sigma = \{0,1\}$

δ:

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

Start state = q_1

$F = \{q_2\}$

```
private String[] states;
private char[] alphabet;
private HashMap<String, HashMap<Character, HashSet<String>>> transitions;
private String startState;
private String[] acceptStates;
public String name;
```

A_{DFA}

Denotes string encoding of some object

Claim: $A_{DFA} = \{\langle B, \omega \rangle \mid B$ is a DFA that accepts string $\omega\}$ is a decidable language.

Language decidability ⟺ Equivalent To ⟺ Computational problem decidability

A_{DFA}

Claim: $A_{DFA} = \{\langle B, \omega \rangle: B$ is a DFA that accepts string $\omega\}$ is a decidable language.

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)

A_{DFA}

Claim: $A_{DFA} = \{\langle B, \omega \rangle : B$ is a DFA that accepts string $\omega\}$ is a decidable language.

Proof:

M_1 = on input $\langle B, \omega \rangle$

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)

A_{DFA}

Claim: $A_{DFA} = \{\langle B, \omega\rangle : B \text{ is a DFA that accepts string } \omega\}$ is a decidable language.

Proof:

M$_1$ = on input $\langle B, \omega\rangle$

1. ?

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)

A_{DFA}

Claim: $A_{DFA} = \{\langle B, \omega\rangle : B$ is a DFA that accepts string $\omega\}$ is a decidable language.

Proof:

M$_1$ = on input $\langle B, \omega\rangle$

1. Run B on ω.
2. ?

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)

A_{DFA}

Claim: $A_{DFA} = \{\langle B, \omega \rangle : B$ is a DFA that accepts string $\omega\}$ is a decidable language.

Proof:

M$_1$ = on input $\langle B, \omega \rangle$

1. Run B on ω.
2. If B accepts, <u>accept</u>. If B 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)

A_{DFA}

Claim: $A_{DFA} = \{\langle B, \omega \rangle : B$ is a DFA that accepts string $\omega\}$ is a decidable language.

Proof:

M$_1$ = on input $\langle B, \omega \rangle$

1. Run B on ω.
2. If B accepts, <u>accept</u>. If B rejects, <u>reject</u>.

M$_1$ is a decider, because ?

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)

A_{DFA}

Claim: $A_{DFA} = \{\langle B, \omega \rangle : B \text{ is a DFA that accepts string } \omega\}$ is a decidable language.

Proof:

M_1 = on input $\langle B, \omega \rangle$

1. Run B on ω.
2. If B accepts, <u>accept</u>. If B rejects, <u>reject</u>.

M_1 is a decider, because all DFAs halt on all input.

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)

A_{DFA}

Claim: $A_{DFA} = \{\langle B, \omega\rangle : B$ is a DFA that accepts string $\omega\}$ is a decidable language.

Proof:

M$_1$ = on input $\langle B, \omega\rangle$

1. Run B on ω.
2. If B accepts, <u>accept</u>. If B rejects, <u>reject</u>.

Implementation Details?

M$_1$ is a decider, because all DFAs halt on all input.

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)

A_{DFA}

Claim: $A_{DFA} = \{\langle B, \omega \rangle : B \text{ is a DFA that accepts string } \omega\}$ is a decidable language.

Proof:

M_1 = on input $\langle B, \omega \rangle$

1. Run B on ω.
2. If B accepts, <u>accept</u>. If B rejects, <u>reject</u>.

Implementation Details?

M_1 is a decider, because all DFAs halt on all input.

States				Transition			Transition					Input String				
q_1	q_2	q_3	#	q_1	0	q_1	q_1	1	q_2	…	#	1	1	0	1	␣

A_{DFA}

Claim: $A_{DFA} = \{\langle B, \omega\rangle : B$ is a DFA that accepts string $\omega\}$ is a decidable language.

Proof:

M$_1$ = on input $\langle B, \omega\rangle$

1. Run B on ω.
2. If B accepts, <u>accept</u>. If B rejects, <u>reject</u>.

Implementation Details?

M$_1$ is a decider, because all DFAs halt on all input.

States | Transition | Transition | Input String

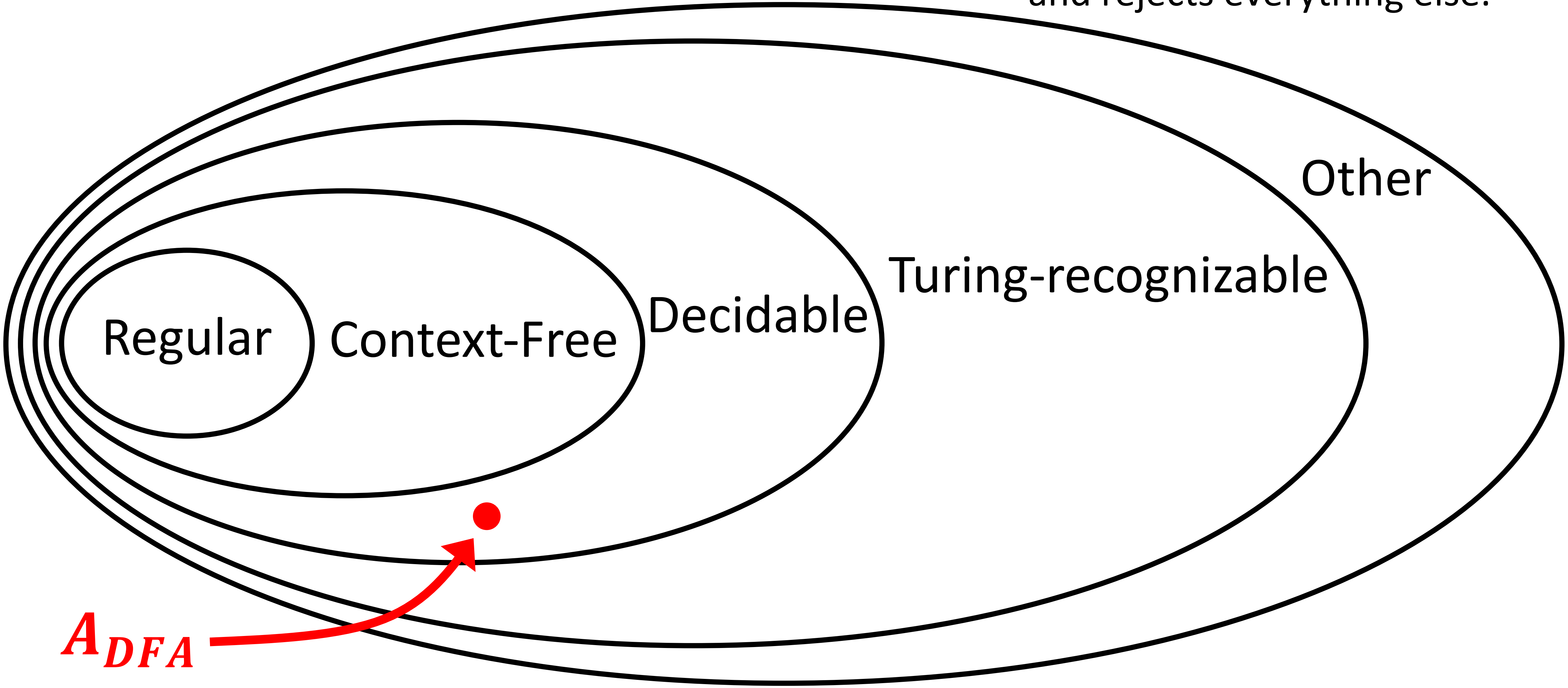
q_1	q_2	q_3	#	q_1	0	q_1	q_1	1	q_2	…	#	1	1	0	1	␣

Mark current state.
Mark current character.
Find applicable transition.
Update state/character.

Computability Hierarchy

Recognizable: ∃ TM that accepts everything in L, and nothing not.

Decidable: ∃ TM that recognizes L and rejects everything else.



A_{NFA}

Claim: $A_{NFA} = \{\langle C, \omega \rangle : C$ is an NFA that accepts string $\omega\}$ is a decidable language.

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)

A_{NFA}

Claim: $A_{NFA} = \{\langle C, \omega\rangle : C \text{ is an NFA that accepts string } \omega\}$ is a decidable language.

Proof:

M$_2$ = on input $\langle C, \omega\rangle$

1. Convert C to an equivalent DFA 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)

A_{NFA}

Claim: $A_{NFA} = \{\langle C, \omega\rangle : C$ is an NFA that accepts string $\omega\}$ is a decidable language.

Proof:

M_2 = on input $\langle C, \omega\rangle$

1. Convert C to an equivalent DFA B.
2. Run M_1 (TM from first example) on $\langle B, \omega\rangle$.

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)

A_{NFA}

Claim: $A_{NFA} = \{\langle C, \omega \rangle : C$ is an NFA that accepts string $\omega\}$ is a decidable language.

Proof:

M_2 = on input $\langle C, \omega \rangle$

1. Convert C to an equivalent DFA B.
2. Run M_1 (TM from first example) on $\langle B, \omega \rangle$.
3. If M_1 accepts, <u>accept</u>. If M_1 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)

A_{NFA}

Claim: $A_{NFA} = \{\langle C, \omega \rangle : C$ is an NFA that accepts string $\omega\}$ is a decidable language.

Proof:

M$_2$ = on input $\langle C, \omega \rangle$

1. Convert C to an equivalent DFA B.
2. Run M$_1$ (TM from first example) on $\langle B, \omega \rangle$.
3. If M$_1$ accepts, <u>accept</u>. If M$_1$ rejects, <u>reject</u>.

M$_2$ is a decider, because ?

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)

A_{NFA}

Claim: $A_{NFA} = \{\langle C, \omega\rangle : C$ is an NFA that accepts string $\omega\}$ is a decidable language.

Proof:

M_2 = on input $\langle C, \omega\rangle$

1. Convert C to an equivalent DFA B.
2. Run M_1 (TM from first example) on $\langle B, \omega\rangle$.
3. If M_1 accepts, <u>accept</u>. If M_1 rejects, <u>reject</u>.

M_2 is a decider, because all DFAs halt on all input.

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)

E_{DFA}

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

E_{DFA}

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

Proof:

M_3 = on input $\langle A \rangle$

1. Mark start state of A.

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

E_{DFA}

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

Proof:

M_3 = on input $\langle A \rangle$

1. Mark start state of A.
2. Mark any state with transition coming from marked state.

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)

E_{DFA}

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

Proof:

M_3 = on input $\langle A \rangle$

1. Mark start state of A.
2. Mark any state with transition coming from marked state.
3. Repeat 2 until no new states are marked.

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)

E_{DFA}

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

Proof:

M$_3$ = on input $\langle A \rangle$

1. Mark start state of A.
2. Mark any state with transition coming from marked state.
3. Repeat 2 until no new states are marked.
4. ???? , <u>accept</u>. Otherwise, <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)

E_{DFA}

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

Proof:

M$_3$ = on input $\langle A \rangle$

1. Mark start state of A.
2. Mark any state with transition coming from marked state.
3. Repeat 2 until no new states are marked.
4. If no accept states are marked, <u>accept</u>. Otherwise, <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)

E_{DFA}

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

Proof:

M$_3$ = on input $\langle A \rangle$

1. Mark start state of A.
2. Mark any state with transition coming from marked state.
3. Repeat 2 until no new states are marked.
4. If no accept states are marked, accept. Otherwise, reject.

M$_3$ is a decider since at least one state must be added for step 2 to repeat, and there are a finite number of states.

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:

?

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?

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)