

# COL 352 Introduction to Automata and Theory of Computation

Nikhil Balaji

Bharti 420  
Indian Institute of Technology, Delhi  
[nbalaji@cse.iitd.ac.in](mailto:nbalaji@cse.iitd.ac.in)

April 5, 2023

Lecture 26: Reductions 1

# Recap

Regular  $\subsetneq$  Context-free  $\subsetneq$  Decidable  $\subsetneq$  Turing Recognizable

Undecidability

# Recap

Regular  $\not\subseteq$  Context-free  $\not\subseteq$  Decidable  $\not\subseteq$  Turing Recognizable

Undecidability

$$A_M = \{\langle M, w \rangle \mid w \in L(M)\}$$

$$E_M = \{\langle M \rangle \mid L(M) = \emptyset\}$$

$$EQ_M = \{\langle M_1, M_2 \rangle \mid L(M_1) = L(M_2)\}$$

# Recap

Regular  $\not\subseteq$  Context-free  $\not\subseteq$  Decidable  $\not\subseteq$  Turing Recognizable

Undecidability

$$A_M = \{\langle M, w \rangle \mid w \in L(M)\}$$

$$E_M = \{\langle M \rangle \mid L(M) = \emptyset\}$$

$$EQ_M = \{\langle M_1, M_2 \rangle \mid L(M_1) = L(M_2)\}$$

- ▶  $A_{TM}$  is undecidable.

# Recap

Regular  $\not\subseteq$  Context-free  $\not\subseteq$  Decidable  $\not\subseteq$  Turing Recognizable

Undecidability

$$A_M = \{\langle M, w \rangle \mid w \in L(M)\}$$

$$E_M = \{\langle M \rangle \mid L(M) = \emptyset\}$$

$$EQ_M = \{\langle M_1, M_2 \rangle \mid L(M_1) = L(M_2)\}$$

- ▶  $A_{TM}$  is undecidable.
- ▶ Universal Turing Machines.

# Reducibility

## *Definition*

A reduction from problem  $P_1$  to problem  $P_2$  is an algorithm to convert instances of a problem  $P_1$  to instances of problem  $P_2$  that have same answers. In this case we say that  $P_2$  is as hard as  $P_1$ .

# Reducibility

## *Definition*

A reduction from problem  $P_1$  to problem  $P_2$  is an algorithm to convert instances of a problem  $P_1$  to instances of problem  $P_2$  that have same answers. In this case we say that  $P_2$  is as hard as  $P_1$ .

## *Theorem*

*If there is a reduction from problem  $P_1$  to problem  $P_2$ , then*

- ▶ *If  $P_1$  is undecidable then so is  $P_2$ .*
- ▶ *If  $P_1$  is non-RE then so is  $P_2$ .*

# Reducibility

## *Definition*

A reduction from problem  $P_1$  to problem  $P_2$  is an algorithm to convert instances of a problem  $P_1$  to instances of problem  $P_2$  that have same answers. In this case we say that  $P_2$  is as hard as  $P_1$ .

## *Theorem*

*If there is a reduction from problem  $P_1$  to problem  $P_2$ , then*

- ▶ *If  $P_1$  is undecidable then so is  $P_2$ .*
- ▶ *If  $P_1$  is non-RE then so is  $P_2$ .*
- ▶ *If  $L$  is decidable then so is  $\overline{L}$ .*



# Reducibility

## *Definition*

A reduction from problem  $P_1$  to problem  $P_2$  is an algorithm to convert instances of a problem  $P_1$  to instances of problem  $P_2$  that have same answers. In this case we say that  $P_2$  is as hard as  $P_1$ .

## *Theorem*

*If there is a reduction from problem  $P_1$  to problem  $P_2$ , then*

- ▶ *If  $P_1$  is undecidable then so is  $P_2$ .*
- ▶ *If  $P_1$  is non-RE then so is  $P_2$ .*
- ▶ If  $L$  is decidable then so is  $\overline{L}$ .
- ▶ If  $L$  and  $\overline{L}$  are Turing recognizable then they are both decidable.

# Other undecidable problems and reducibility

# Other undecidable problems and reducibility

Reducing  $A_{TM}$  to another problem to prove undecidability.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

# Other undecidable problems and reducibility

Reducing  $A_{TM}$  to another problem to prove undecidability.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

# Other undecidable problems and reducibility

Reducing  $A_{TM}$  to another problem to prove undecidability.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable.

# Other undecidable problems and reducibility

Reducing  $A_{TM}$  to another problem to prove undecidability.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let  $\mathcal{H}$  be the TM deciding Halt.

# Other undecidable problems and reducibility

Reducing  $A_{TM}$  to another problem to prove undecidability.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let  $\mathcal{H}$  be the TM deciding Halt.

$\mathcal{A}$ : Run  $\mathcal{H}$  on  $(M, w)$ .

# Other undecidable problems and reducibility

Reducing  $A_{TM}$  to another problem to prove undecidability.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let  $\mathcal{H}$  be the TM deciding Halt.

$\mathcal{A}$ : Run  $\mathcal{H}$  on  $(M, w)$ . If it rejects then reject,



# Other undecidable problems and reducibility

Reducing  $A_{TM}$  to another problem to prove undecidability.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let  $\mathcal{H}$  be the TM deciding Halt.

$\mathcal{A}$ : Run  $\mathcal{H}$  on  $(M, w)$ . If it rejects then reject, else do as per  $M$  on  $w$ .

# Other undecidable problems and reducibility

Reducing  $A_{TM}$  to another problem to prove undecidability.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let  $\mathcal{H}$  be the TM deciding Halt.

$\mathcal{A}$ : Run  $\mathcal{H}$  on  $(M, w)$ . If it rejects then reject, else do as per  $M$  on  $w$ .

$\mathcal{A}$  accepts  $(M, w)$  if  $M$  accepts  $w$

# Other undecidable problems and reducibility

Reducing  $A_{TM}$  to another problem to prove undecidability.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let  $\mathcal{H}$  be the TM deciding Halt.

$\mathcal{A}$ : Run  $\mathcal{H}$  on  $(M, w)$ . If it rejects then reject, else do as per  $M$  on  $w$ .

$\mathcal{A}$  accepts  $(M, w)$  if  $M$  accepts  $w$  and rejects it if either  $M$  rejects  $w$

## Other undecidable problems and reducibility

Reducing  $A_{TM}$  to another problem to prove undecidability.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let  $\mathcal{H}$  be the TM deciding Halt.

$\mathcal{A}$ : Run  $\mathcal{H}$  on  $(M, w)$ . If it rejects then reject, else do as per  $M$  on  $w$ .

$\mathcal{A}$  accepts  $(M, w)$  if  $M$  accepts  $w$  and rejects it if either  $M$  rejects  $w$  or  $M$  loops forever on  $w$ .

## Other undecidable problems and reducibility

Reducing  $A_{TM}$  to another problem to prove undecidability.

$$\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$$

We would like to show that Halt is undecidable.

Assume that Halt is decidable. Let  $\mathcal{H}$  be the TM deciding Halt.

$\mathcal{A}$ : Run  $\mathcal{H}$  on  $(M, w)$ . If it rejects then reject, else do as per  $M$  on  $w$ .

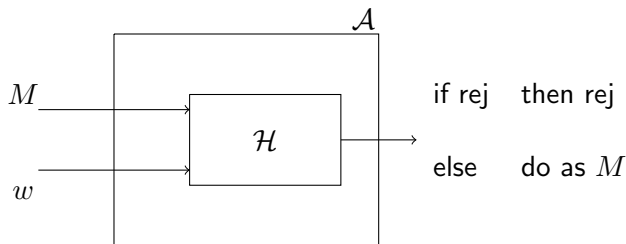
$\mathcal{A}$  accepts  $(M, w)$  if  $M$  accepts  $w$  and rejects it if either  $M$  rejects  $w$  or  $M$  loops forever on  $w$ .

$\mathcal{H}$  decides Halt if and only if  $\mathcal{A}$  decides  $A_{TM}$ .

# The halting problem

## *Lemma*

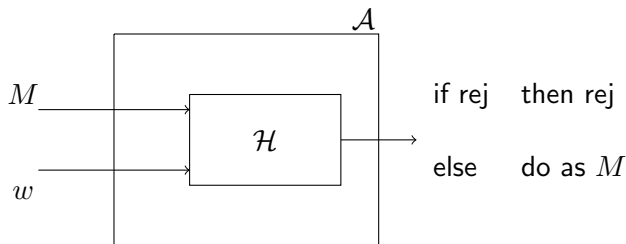
*The halting problem,  $\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$ , is undecidable.*



# The halting problem

## *Lemma*

*The halting problem,  $\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$ , is undecidable.*

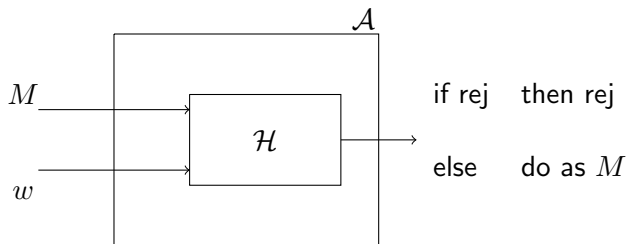


If Halt is decidable then  $\mathcal{A}$  decides  $A_{TM}$

# The halting problem

## *Lemma*

*The halting problem,  $\text{Halt} = \{(M, w) \mid M \text{ halts on } w\}$ , is undecidable.*



If Halt is decidable then  $\mathcal{A}$  decides  $A_{TM}$ , which is a contradiction.



# Emptiness problem for TM

## *Lemma*

*The emptiness problem for TMs,  $E_{TM} = \{\langle M \rangle \mid L(M) = \emptyset\}$ , is undecidable.*

Assume for the sake of contradiction that it is decidable.

# Emptiness problem for TM

## *Lemma*

*The emptiness problem for TMs,  $E_{TM} = \{\langle M \rangle \mid L(M) = \emptyset\}$ , is undecidable.*

Assume for the sake of contradiction that it is decidable. Let  $T$  be a machine that decides  $E_{TM}$ .

# Emptiness problem for TM

## *Lemma*

*The emptiness problem for TMs,  $E_{TM} = \{\langle M \rangle \mid L(M) = \emptyset\}$ , is undecidable.*

Assume for the sake of contradiction that it is decidable. Let  $T$  be a machine that decides  $E_{TM}$ .

Let  $T'_{M,w}$  be as follows:

# Emptiness problem for TM

## *Lemma*

*The emptiness problem for TMs,  $E_{TM} = \{\langle M \rangle \mid L(M) = \emptyset\}$ , is undecidable.*

Assume for the sake of contradiction that it is decidable. Let  $T$  be a machine that decides  $E_{TM}$ .

Let  $T'_{M,w}$  be as follows:

On input  $x$

{  
if  $w \neq x$  then reject  
else do as per  $M$   
}

# Emptiness problem for TM

## *Lemma*

*The emptiness problem for TMs,  $E_{TM} = \{\langle M \rangle \mid L(M) = \emptyset\}$ , is undecidable.*

Assume for the sake of contradiction that it is decidable. Let  $T$  be a machine that decides  $E_{TM}$ .

Let  $T'_{M,w}$  be as follows:

On input  $x$

{  
if  $w \neq x$  then reject  
else do as per  $M$   
}

$$L(T'_{M,w}) = \begin{cases} \{w\} & \text{if } M \text{ acc } w \\ \emptyset & \text{otherwise} \end{cases}$$

# Emptiness problem for TM

## Lemma

The emptiness problem for TMs,  $E_{TM} = \{\langle M \rangle \mid L(M) = \emptyset\}$ , is undecidable.

Assume for the sake of contradiction that it is decidable. Let  $T$  be a machine that decides  $E_{TM}$ .

Let  $T'_{M,w}$  be as follows:

Let  $A$  be as follows:

On input  $x$

{  
if  $w \neq x$  then reject  
else do as per  $M$   
}

$$L(T'_{M,w}) = \begin{cases} \{w\} & \text{if } M \text{ acc } w \\ \emptyset & \text{otherwise} \end{cases}$$

# Emptiness problem for TM

## Lemma

The emptiness problem for TMs,  $E_{TM} = \{\langle M \rangle \mid L(M) = \emptyset\}$ , is undecidable.

Assume for the sake of contradiction that it is decidable. Let  $T$  be a machine that decides  $E_{TM}$ .

Let  $T'_{M,w}$  be as follows:

On input  $x$

{  
if  $w \neq x$  then reject  
else do as per  $M$   
}

$$L(T'_{M,w}) = \begin{cases} \{w\} & \text{if } M \text{ acc } w \\ \emptyset & \text{otherwise} \end{cases}$$

Let  $A$  be as follows:

On input  $M, w$

{  
Create machine  $T'_{M,w}$ .  
If  $T$  on  $\langle T'_{M,w} \rangle$  rejects  
then accept  
else reject  
}

This shows that if  $E_{TM}$  is decidable then  $A_{TM}$  is decidable.

# Equality for TM

## *Lemma*

*The equality problem for TMs,  $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$ , is undecidable.*



# Equality for TM

## *Lemma*

*The equality problem for TMs,  $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$ , is undecidable.*

Assume for the sake of contradiction that  $EQ_{TM}$  is decidable. Let  $M$  be the TM for it.

# Equality for TM

## *Lemma*

*The equality problem for TMs,  $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$ , is undecidable.*

Assume for the sake of contradiction that  $EQ_{TM}$  is decidable. Let  $M$  be the TM for it.

Let  $M_1$  be a machine that rejects all strings. That is,  $L(M_1) = \emptyset$ .

# Equality for TM

## *Lemma*

*The equality problem for TMs,  $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$ , is undecidable.*

Assume for the sake of contradiction that  $EQ_{TM}$  is decidable. Let  $M$  be the TM for it.

Let  $M_1$  be a machine that rejects all strings. That is,  $L(M_1) = \emptyset$ .

Given a machine  $M_2$  as an input, use  $M$  to check whether  $L(M_2) = L(M_1)$

# Equality for TM

## *Lemma*

*The equality problem for TMs,  $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$ , is undecidable.*

Assume for the sake of contradiction that  $EQ_{TM}$  is decidable. Let  $M$  be the TM for it.

Let  $M_1$  be a machine that rejects all strings. That is,  $L(M_1) = \emptyset$ .

Given a machine  $M_2$  as an input, use  $M$  to check whether  $L(M_2) = L(M_1)$ , i.e. to check whether  $L(M_2) = \emptyset$  or not.

# Equality for TM

## *Lemma*

*The equality problem for TMs,  $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$ , is undecidable.*

Assume for the sake of contradiction that  $EQ_{TM}$  is decidable. Let  $M$  be the TM for it.

Let  $M_1$  be a machine that rejects all strings. That is,  $L(M_1) = \emptyset$ .

Given a machine  $M_2$  as an input, use  $M$  to check whether  $L(M_2) = L(M_1)$ , i.e. to check whether  $L(M_2) = \emptyset$  or not.

This implies that if  $EQ_{TM}$  is decidable then  $E_{TM}$  is decidable.

# Equality for TM

## *Lemma*

*The equality problem for TMs,  $EQ_{TM} = \{(M_1, M_2) \mid L(M_1) = L(M_2)\}$ , is undecidable.*

Assume for the sake of contradiction that  $EQ_{TM}$  is decidable. Let  $M$  be the TM for it.

Let  $M_1$  be a machine that rejects all strings. That is,  $L(M_1) = \emptyset$ .

Given a machine  $M_2$  as an input, use  $M$  to check whether  $L(M_2) = L(M_1)$ , i.e. to check whether  $L(M_2) = \emptyset$  or not.

This implies that if  $EQ_{TM}$  is decidable then  $E_{TM}$  is decidable.

But from the previous result we know that  $E_{TM}$  is undecidable.

# Regularity checking

*Lemma*

$REG_{TM} = \{\langle M \rangle \mid L(M) \text{ is regular}\}$  is undecidable.

# Regularity checking

## *Lemma*

$REG_{TM} = \{\langle M \rangle \mid L(M) \text{ is regular}\}$  is undecidable.

Assume for the sake of contradiction that a TM  $R$  is a TM that decides  $REG_{TM}$ .



# Regularity checking

## *Lemma*

$REG_{TM} = \{\langle M \rangle \mid L(M) \text{ is regular}\}$  is undecidable.

Assume for the sake of contradiction that a TM  $R$  is a TM that decides  $REG_{TM}$ .

Let  $R'_{M,w}$  be s.t.

$$L(R'_{M,w}) = \begin{cases} \{0^n 1^n \mid n \geq 0\} & \text{if } M \text{ rej } w \\ \Sigma^* & \text{if } M \text{ acc } w \end{cases}$$

# Regularity checking

## *Lemma*

$REG_{TM} = \{\langle M \rangle \mid L(M) \text{ is regular}\}$  is undecidable.

Assume for the sake of contradiction that a TM  $R$  is a TM that decides  $REG_{TM}$ .

Let  $R'_{M,w}$  be s.t.

$$L(R'_{M,w}) = \begin{cases} \{0^n 1^n \mid n \geq 0\} & \text{if } M \text{ rej } w \\ \Sigma^* & \text{if } M \text{ acc } w \end{cases}$$

If we get such an  $R'_{M,w}$  we can design  $A$  as follows.

# Regularity checking

## *Lemma*

$REG_{TM} = \{\langle M \rangle \mid L(M) \text{ is regular}\}$  is undecidable.

Assume for the sake of contradiction that a TM  $R$  is a TM that decides  $REG_{TM}$ .

Let  $R'_{M,w}$  be s.t.

$$L(R'_{M,w}) = \begin{cases} \{0^n 1^n \mid n \geq 0\} & \text{if } M \text{ rej } w \\ \Sigma^* & \text{if } M \text{ acc } w \end{cases}$$

If we get such an  $R'_{M,w}$  we can design  $A$  as follows.

# Regularity checking

## *Lemma*

$REG_{TM} = \{\langle M \rangle \mid L(M) \text{ is regular}\}$  is undecidable.

Assume for the sake of contradiction that a TM  $R$  is a TM that decides  $REG_{TM}$ .

Let  $R'_{M,w}$  be s.t.

Let  $A$  be as follows:

$$L(R'_{M,w}) = \begin{cases} \{0^n 1^n \mid n \geq 0\} & \text{if } M \text{ rej } w \\ \Sigma^* & \text{if } M \text{ acc } w \end{cases}$$

If we get such an  $R'_{M,w}$  we can design  $A$  as a follows.

# Regularity checking

## Lemma

$REG_{TM} = \{\langle M \rangle \mid L(M) \text{ is regular}\}$  is undecidable.

Assume for the sake of contradiction that a TM  $R$  is a TM that decides  $REG_{TM}$ .

Let  $R'_{M,w}$  be s.t.

$$L(R'_{M,w}) = \begin{cases} \{0^n 1^n \mid n \geq 0\} & \text{if } M \text{ rej } w \\ \Sigma^* & \text{if } M \text{ acc } w \end{cases}$$

If we get such an  $R'_{M,w}$  we can design  $A$  as follows.

Let  $A$  be as follows:

On input  $M, w$

{  
Create machine  $R'_{M,w}$ .  
If  $R$  on  $\langle R'_{M,w} \rangle$  accepts  
then accept  
else reject  
}

# Regularity checking

*Lemma*

$REG_{TM} = \{\langle M \rangle \mid L(M) \text{ is regular}\}$  is undecidable.

# Regularity checking

## *Lemma*

$REG_{TM} = \{\langle M \rangle \mid L(M) \text{ is regular}\}$  is undecidable.

Assume for the sake of contradiction that a TM  $R$  be a TM that decides  $REG_{TM}$ .

Let  $R'_{M,w}$  be as follows:

# Regularity checking

## *Lemma*

$REG_{TM} = \{\langle M \rangle \mid L(M) \text{ is regular}\}$  is undecidable.

Assume for the sake of contradiction that a TM  $R$  be a TM that decides  $REG_{TM}$ .

Let  $R'_{M,w}$  be as follows:

On input  $x$

{

if  $x = 0^n 1^n$

then accept

else run  $M$  on  $w$  and

if  $M$  acc  $w$  then acc

else rej

}



# Regularity checking

## *Lemma*

$REG_{TM} = \{\langle M \rangle \mid L(M) \text{ is regular}\}$  is undecidable.

Assume for the sake of contradiction that a TM  $R$  be a TM that decides  $REG_{TM}$ .

Let  $R'_{M,w}$  be as follows:

Let  $A$  be as follows:

On input  $x$

{

if  $x = 0^n 1^n$

then accept

else run  $M$  on  $w$  and

if  $M$  acc  $w$  then acc

else rej

}

# Regularity checking

## *Lemma*

$REG_{TM} = \{\langle M \rangle \mid L(M) \text{ is regular}\}$  is undecidable.

Assume for the sake of contradiction that a TM  $R$  be a TM that decides  $REG_{TM}$ .

Let  $R'_{M,w}$  be as follows:

On input  $x$

```
{
  if  $x = 0^n 1^n$ 
  then accept
  else run  $M$  on  $w$  and
    if  $M$  acc  $w$  then acc
    else rej
}
```

Let  $A$  be as follows:

On input  $M, w$

```
{
  Create machine  $R'_{M,w}$ .
  If  $R$  on  $\langle R'_{M,w} \rangle$  accepts
  then accept
  else reject
}
```