Algorithms & Models of Computation CS/ECE 374, Fall 2020

9.3

The halting theorem

Encodings

```
M: Turing machine
```

 $\langle M \rangle$: a binary string uniquely describing M (i.e., it is a number.

w: An input string.

 $\langle M, w \rangle$: A unique binary string encoding both M and input w.

$$\mathbf{A}_{\mathrm{TM}} = \left\{ \langle extbf{ extit{M}}, extbf{ extit{w}}
angle \; | \; extbf{ extit{M}} \; ext{is a TM and } extbf{ extit{M}} \; ext{accepts } extbf{ extit{w}}
ight\}.$$

Encodings

```
M: Turing machine
```

 $\langle M \rangle$: a binary string uniquely describing M (i.e., it is a number.

w: An input string.

 $\langle M, w \rangle$: A unique binary string encoding both M and input w.

$$\mathbf{A}_{\mathrm{TM}} = \left\{ \langle extbf{ extit{M}}, extbf{ extit{w}}
angle \; | \; extbf{ extit{M}} \; ext{is a TM} \; ext{and} \; extbf{ extit{M}} \; ext{accepts} \; extbf{ extit{w}}
ight\}.$$

Encodings

M: Turing machine

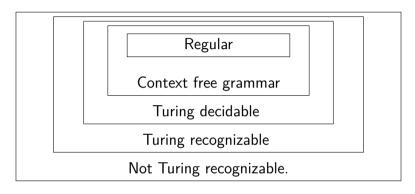
 $\langle M \rangle$: a binary string uniquely describing M (i.e., it is a number.

w: An input string.

 $\langle M, w \rangle$: A unique binary string encoding both M and input w.

$$\mathbf{A}_{\mathrm{TM}} = \left\{ \langle extbf{ extit{M}}, extbf{ extit{w}}
angle \; | \; extbf{ extit{M}} \; ext{is a TM} \; ext{and} \; extbf{ extit{M}} \; ext{accepts} \; extbf{ extit{w}}
ight\}.$$

Complexity classes



A_{TM} is TM recognizable...

$$\mathbf{A}_{\mathrm{TM}} = \left\{ \langle extbf{ extit{M}}, extbf{ extit{w}}
angle \; egin{aligned} extbf{ extit{M}} & extbf{ extit{a}} & extbf{ extit{TM}} \end{aligned}
ight.$$
 and $extbf{ extit{M}}$ accepts $extbf{ extit{w}}
ight\}$.

Lemma

 \mathbf{A}_{TM} is Turing recognizable.

Proof

Input: $\langle M, w \rangle$.

Using UTM simulate running M on w. If M accepts w then accept, if M rejects then reject. Otherwise, the simulation runs forever.

A_{TM} is TM recognizable...

$$\mathbf{A}_{\mathrm{TM}} = \left\{ \langle extbf{ extit{M}}, extbf{ extit{w}}
angle \; | \; extbf{ extit{M}} \; ext{is a TM} \; ext{and} \; extbf{ extit{M}} \; ext{accepts} \; extbf{ extit{w}}
ight\}.$$

Lemma

A_{TM} is Turing recognizable.

Proof.

Input: $\langle M, w \rangle$.

Using \overline{UTM} simulate running M on w. If M accepts w then accept, if M rejects then reject. Otherwise, the simulation runs forever.

A_{TM} is not TM decidable!

$$\mathbf{A}_{\mathrm{TM}} = \left\{ \langle m{M}, m{w}
angle \; igg| \; m{M} \; ext{is a TM} \; ext{and} \; m{M} \; ext{accepts} \; m{w}
ight\}.$$

Theorem (The halting theorem.)

 \mathbf{A}_{TM} is not Turing decidable.

Proof: Assume **A**_{TM} is TM decidable...

Halt: TM deciding A_{TM} . **Halt** always halts, and works as follows:

$$\mathsf{Halt}\Big(\langle M, w \rangle\Big) = \begin{cases} \mathsf{accept} & \textit{M} \; \mathsf{accepts} \; \textit{w} \\ \mathsf{reject} & \textit{M} \; \mathsf{does} \; \mathsf{not} \; \mathsf{accept} \; \textit{w}. \end{cases}$$

A_{TM} is not TM decidable!

$$\mathbf{A}_{\mathrm{TM}} = \left\{ \langle extbf{ extit{M}}, extbf{ extit{w}}
angle \; igg| \; extbf{ extit{M}} ext{ is a TM and } extbf{ extit{M}} ext{ accepts } extbf{ extit{w}}
ight\}.$$

Theorem (The halting theorem.)

 \mathbf{A}_{TM} is not Turing decidable.

Proof: Assume A_{TM} is TM decidable...

Halt: TM deciding A_{TM} . **Halt** always halts, and works as follows:

$$\mathsf{Halt}\Big(\langle M, w \rangle\Big) = egin{cases} \mathsf{accept} & M \; \mathsf{accepts} \; w \ \mathsf{reject} & M \; \mathsf{does} \; \mathsf{not} \; \mathsf{accept} \; w. \end{cases}$$

A_{TM} is not TM decidable!

$$\mathbf{A}_{\mathrm{TM}} = \left\{ \langle extbf{ extit{M}}, extbf{ extit{w}}
angle \; | \; extbf{ extit{M}} \; ext{is a TM} \; ext{and} \; extbf{ extit{M}} \; ext{accepts} \; extbf{ extit{w}}
ight\}.$$

Theorem (The halting theorem.)

 \mathbf{A}_{TM} is not Turing decidable.

Proof: Assume **A**_{TM} is TM decidable...

Halt: TM deciding A_{TM} . Halt always halts, and works as follows:

$$\mathsf{Halt}\Big(\langle M, w \rangle\Big) = egin{cases} \mathsf{accept} & M \ \mathsf{accepts} \ w \ \mathsf{reject} & M \ \mathsf{does} \ \mathsf{not} \ \mathsf{accept} \ w. \end{cases}$$

We build the following new function:

```
Flipper(\langle M \rangle)
res \leftarrow Halt(\langle M, M \rangle)
if res is accept then
reject
else
accept
```

Flipper always stops:

$$\mathsf{Flipper}\Big(\langle \mathbf{M} \rangle \Big) = \begin{cases} \mathsf{reject} & \mathbf{M} \; \mathsf{accepts} \; \langle \mathbf{M} \rangle \\ \mathsf{accept} & \mathbf{M} \; \mathsf{does} \; \mathsf{not} \; \mathsf{accept} \; \langle \mathbf{M} \rangle \,. \end{cases}$$

We build the following new function:

```
Flipper(\langle M \rangle)
res \leftarrow Halt(\langle M, M \rangle)
if res is accept then
reject
else
accept
```

Flipper always stops:

Flipper is a TM (duh!), and as such it has an encoding \langle **Flipper** \rangle . Run **Flipper** on itself:

$$\begin{aligned} & \textbf{Flipper}\Big(\, \langle \textbf{Flipper} \rangle \Big) = \begin{cases} \text{reject} & \textbf{Flipper} \text{ accepts } \langle \textbf{Flipper} \rangle \\ \text{accept} & \textbf{Flipper} \text{ does not accept } \langle \textbf{Flipper} \rangle \,. \end{cases} \end{aligned}$$

This is absurd. Ridiculous even! Assumption that **Halt** exists is false. \implies $A_{\rm TM}$ is not ${\rm TM}$ decidable.

Har-Peled (UIUC) CS374 20 Fall 2020 20 / 33

Flipper is a TM (duh!), and as such it has an encoding \langle **Flipper** \rangle . Run **Flipper** on itself:

This is absurd. Ridiculous even!

Assumption that **Halt** exists is false. \implies $A_{\rm TM}$ is not ${\rm TM}$ decidable.

Har-Peled (UIUC) CS374 20 Fall 2020 20 / 33

Flipper is a TM (duh!), and as such it has an encoding \langle **Flipper** \rangle . Run **Flipper** on itself:

This is absurd. Ridiculous even!

Assumption that **Halt** exists is false. \implies A_{TM} is not TM decidable.

Har-Peled (UIUC) CS374 20 Fall 2020 20 / 33

But where is the diagonalization argument?????

	$\langle \pmb{M}_1 angle$	$\langle M_2 angle$	$\langle M_3 \rangle$	$\langle M_4 angle$	
M_1	rej	acc	rej	rej	
M_1 M_2 M_3 M_4	rej	acc	rej	acc	
M_3	acc	acc	acc	rej	
M_4	rej	acc	acc	rej	
:	:	:	:	:	$\langle \cdot, \cdot \rangle$

THE END

...

(for now)