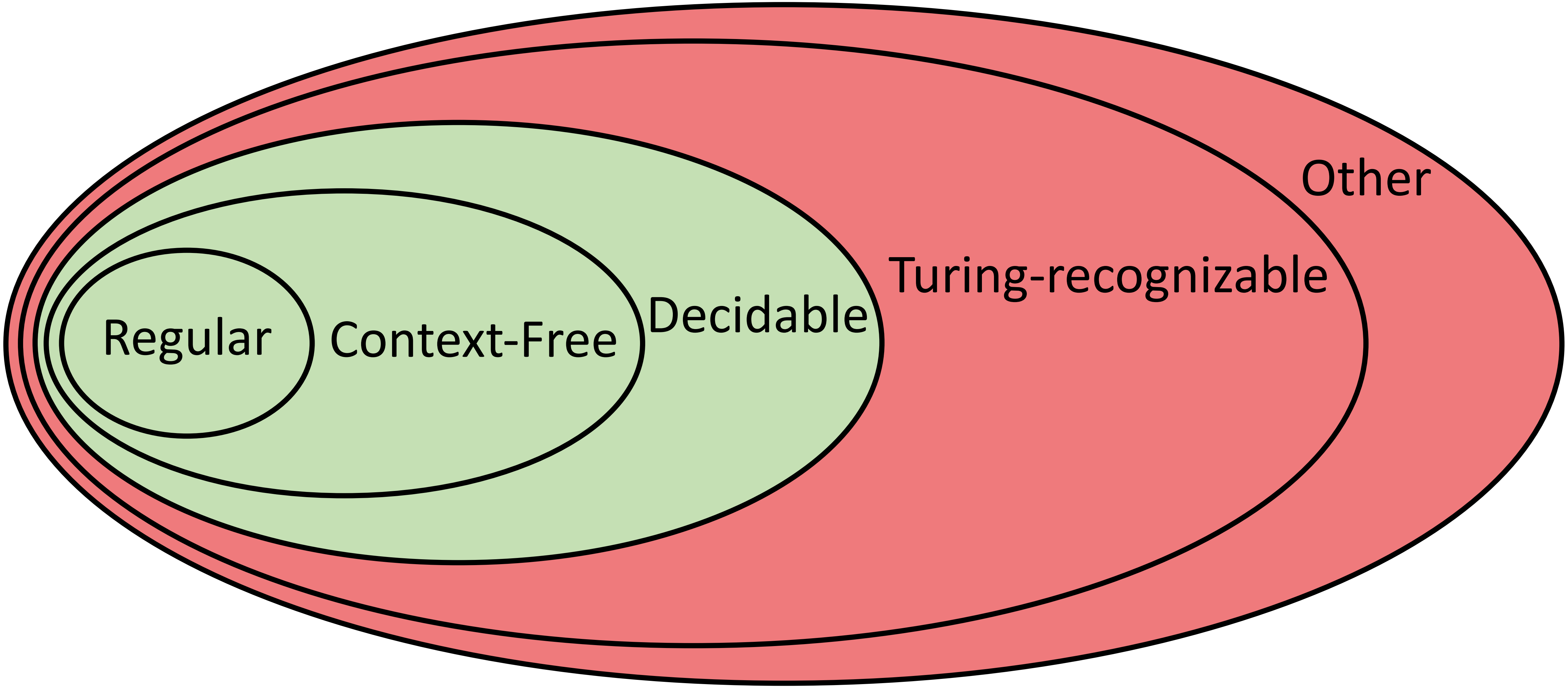
Undecidability
CSCI 338

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

A_{TM}

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

Proof:

M$_5$ = on input $\langle M, \omega \rangle$

1. Run M on ω.
2. If M accepts, <u>accept</u>. If M 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_{TM}

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

Proof:

M$_5$ = on input $\langle M, \omega\rangle$

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

What if M loops (doesn't terminate)?

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_{TM}

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

Proof:

M_S = on input $\langle M, \omega \rangle$

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

Not a Decider!!!

What if M loops (doesn't terminate)?

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_{TM}

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

Proof: ?

A_{TM}

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

Proof: Suppose A_{TM} is decidable.

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega\rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega \rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

Make a new TM D:

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega\rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

Make a new TM D:

D = on input $\langle N\rangle$, for TM N

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega \rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

Make a new TM D:

D = on input $\langle N \rangle$, for TM N

1. Run H on $\langle N, \langle N \rangle \rangle$.

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega \rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

Make a new TM D:

D = on input $\langle N \rangle$, for TM N

1. Run H on $\langle N, \langle N \rangle \rangle$.

TM N

String encoding of N

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega\rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

Make a new TM D:

D = on input $\langle N\rangle$, for TM N

1. Run H on $\langle N, \langle N\rangle\rangle$.
2. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega\rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

Make a new TM D:

D = on input $\langle N\rangle$, for TM N

1. Run H on $\langle N, \langle N\rangle\rangle$.
2. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

What happens with $D(\langle D\rangle)$?

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega \rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

Make a new TM D:

D = on input $\langle N \rangle$, for TM N

1. Run H on $\langle N, \langle N \rangle \rangle$.
2. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

What happens with $D(\langle D \rangle)$? $D(\langle D \rangle) = \begin{cases} \text{accept, if ?} \\ \text{reject, if ?} \end{cases}$

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega \rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

Make a new TM D:

D = on input $\langle N \rangle$, for TM N

1. Run H on $\langle N, \langle N \rangle \rangle$.
2. If H accepts, reject. **If H rejects, accept.**

What happens with $D(\langle D \rangle)$? $D(\langle D \rangle) = \begin{cases} \textbf{accept}\text{, if ?} \\ \text{reject, if ?} \end{cases}$

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega \rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \textbf{reject, if } \boldsymbol{M} \textbf{ does not accept } \boldsymbol{\omega} \end{cases}$$

Make a new TM D:

D = on input $\langle N \rangle$, for TM N

1. Run H on $\langle N, \langle N \rangle \rangle$.
2. If H accepts, <u>reject</u>. **If H rejects, <u>accept</u>**.

What happens with $D(\langle D \rangle)$? $D(\langle D \rangle) = \begin{cases} \textbf{accept}\text{, if ?} \\ \text{reject, if ?} \end{cases}$

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega \rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \textbf{reject, if } \boldsymbol{M} \textbf{ does not accept } \boldsymbol{\omega} \end{cases}$$

Make a new TM D:

D = on input $\langle N \rangle$, for TM N

1. Run H on $\langle N, \langle N \rangle \rangle$.
2. If H accepts, <u>reject</u>. **If H rejects, <u>accept</u>**.

What happens with $D(\langle D \rangle)$? $D(\langle D \rangle) = \begin{cases} \textbf{accept}\text{, if } D \text{ does not accept } \langle D \rangle \\ \text{reject, if ?} \end{cases}$

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega \rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

Make a new TM D:

D = on input $\langle N \rangle$, for TM N

1. Run H on $\langle N, \langle N \rangle \rangle$.
2. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

What happens with $D(\langle D \rangle)$? $D(\langle D \rangle) = \begin{cases} \text{accept, if } D \text{ does not accept } \langle D \rangle \\ \text{reject, if } D \text{ accepts } \langle D \rangle \end{cases}$

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega \rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

Make a new TM D:

D = on input $\langle N \rangle$, for TM N

1. Run H on $\langle N, \langle N \rangle \rangle$.
2. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

What happens with $D(\langle D \rangle)$? $D(\langle D \rangle) = \begin{cases} \text{accept, if } D \text{ does not accept } \langle D \rangle \\ \text{reject, if } D \text{ accepts } \langle D \rangle \end{cases}$

D accepts $\langle D \rangle$, so long as D does not accept $\langle D \rangle$.

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega \rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

Make a new TM D:

D = on input $\langle N \rangle$, for TM N

1. Run H on $\langle N, \langle N \rangle \rangle$.
2. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

What happens with $D(\langle D \rangle)$? $D(\langle D \rangle) = \begin{cases} \text{accept, if } D \text{ does not accept } \langle D \rangle \\ \text{reject, if } D \text{ accepts } \langle D \rangle \end{cases}$

D accepts $\langle D \rangle$, so long as D does not accept $\langle D \rangle$.

$\Rightarrow$ TM D cannot exist $\Longrightarrow$ TM H cannot exist $\Longrightarrow$ A_{TM} is undecidable

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega \rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

Make a new TM D:

D =

1

2

No algorithm can determine (with a 'yes' or 'no') whether or not an algorithm or program will accept some input.

What happens with $D(\langle D \rangle)$? $D(\langle D \rangle) = \begin{cases} \text{accept, if } D \text{ does not accept } \langle D \rangle \\ \text{reject, if } D \text{ accepts } \langle D \rangle \end{cases}$

D accepts $\langle D \rangle$, so long as D does not accept $\langle D \rangle$.

$\Rightarrow$ TM D cannot exist $\Longrightarrow$ TM H cannot exist $\Longrightarrow$ A_{TM} is undecidable

Set of Turing Machines

Claim: The set of all possible Turing machines is countable.

Proof: ?

I.e. You can list all Turing machines in an ordered list.

Set of Turing Machines

Claim: The set of all possible Turing machines is countable.

Proof: Encode each TM as a binary string and sort in lexicographic order.

A_{TM}

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

Proof: Suppose A_{TM} is decidable. Let TM H be its decider:

$$H(\langle M, \omega \rangle) = \begin{cases} \text{accept, if } M \text{ accepts } \omega \\ \text{reject, if } M \text{ does not accept } \omega \end{cases}$$

Make a new TM D:

D = on input $\langle N \rangle$, for TM N

1. Run H on $\langle N, \langle N \rangle \rangle$.
2. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

What happens with $D(\langle D \rangle)$? $D(\langle D \rangle) = \begin{cases} \text{accept, if } D \text{ does not accept } \langle D \rangle \\ \text{reject, if } D \text{ accepts } \langle D \rangle \end{cases}$

D accepts $\langle D \rangle$, so long as D does not accept $\langle D \rangle$.

$\Rightarrow$ TM D cannot exist $\Longrightarrow$ TM H cannot exist $\Longrightarrow$ A_{TM} is undecidable

Set of Turing Machines

Consider each TM M and its string representation $\langle M \rangle$:

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$
M_1					
M_2					
M_3					
M_4					
$\vdots$					

Set of Turing Machines

Consider each TM M and its string representation $\langle M \rangle$:

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$
M_1	r	a	r	r	
M_2	a	r	r	a	
M_3	r	a	a	r	
M_4	r	r	a	r	$\cdots$
$\vdots$			$\vdots$		

Result of running H on $\langle M_2, \langle M_4 \rangle \rangle$

H = Decider for A_{TM}

Set of Turing Machines

Consider each TM M and its string representation $\langle M \rangle$:

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$	$\langle D \rangle$	$\cdots$
M_1	r	a	r	r		a	
M_2	a	r	r	a		r	
M_3	r	a	a	r		a	
M_4	r	r	a	r	$\cdots$	a	$\cdots$
$\vdots$			$\vdots$		$\ddots$		
D	?	?	?	?		?	
$\vdots$			$\vdots$				$\ddots$

H = Decider for A_{TM}

D = on input $\langle N \rangle$, for TM N
1. Run H on $\langle N, \langle N \rangle \rangle$.
2. If H accepts, reject.
   If H rejects, accept.

Set of Turing Machines

Consider each TM M and its string representation $\langle M \rangle$:

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$	$\langle D \rangle$	$\cdots$
M_1	r	a	r	r		a	
M_2	a	r	r	a		r	
M_3	r	a	a	r		a	
M_4	r	r	a	r	$\cdots$	a	$\cdots$
$\vdots$			$\vdots$		$\ddots$		
D	a	?	?	?		?	
$\vdots$			$\vdots$				$\ddots$

H = Decider for A_{TM}

D = on input $\langle N \rangle$, for TM N

1. Run H on $\langle N, \langle N \rangle \rangle$.
2. If H accepts, <u>reject</u>. If H rejects, <u>accept</u>.

Set of Turing Machines

Consider each TM M and its string representation $\langle M \rangle$:

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$	$\langle D \rangle$	$\cdots$
M_1	r	a	r	r		a	
M_2	a	r	r	a		r	
M_3	r	a	a	r		a	
M_4	r	r	a	r	$\cdots$	a	$\cdots$
$\vdots$			$\vdots$		$\ddots$		
D	a	a	r	a		?	
$\vdots$			$\vdots$				$\ddots$

H = Decider for A_{TM}

D = on input $\langle N \rangle$, for TM N

1. Run H on $\langle N, \langle N \rangle \rangle$.
2. If H accepts, reject.
   If H rejects, accept.

Set of Turing Machines

Consider each TM M and its string representation $\langle M \rangle$:

	$\langle M_1 \rangle$	$\langle M_2 \rangle$	$\langle M_3 \rangle$	$\langle M_4 \rangle$	$\cdots$	$\langle D \rangle$	$\cdots$
M_1	r	a	r	r		a	
M_2	a	r	r	a		r	
M_3	r	a	a	r		a	
M_4	r	r	a	r	$\cdots$	a	$\cdots$
$\vdots$			$\vdots$		$\ddots$		
D	a	a	r	a		?	
$\vdots$			$\vdots$				$\ddots$

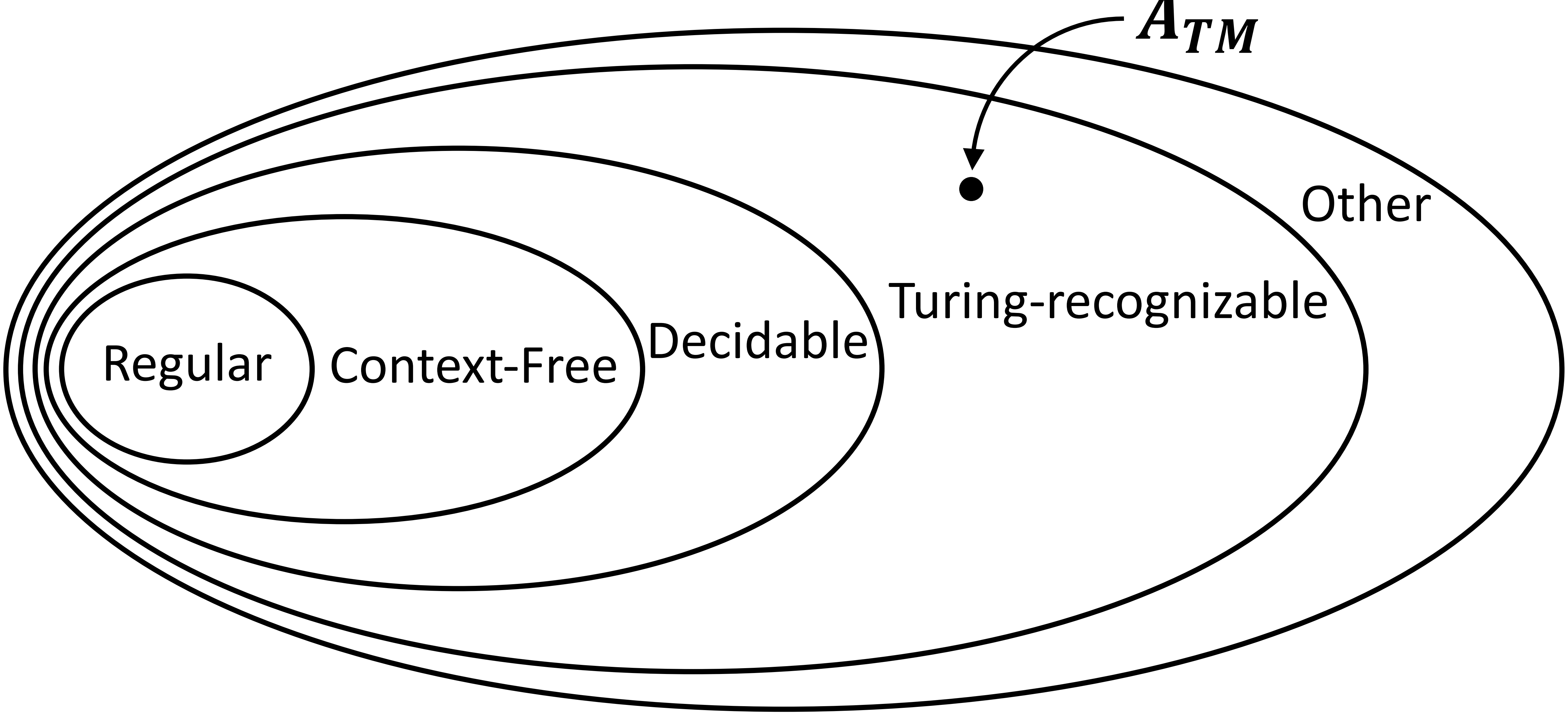
D is a TM, so it must be in the list and H is a decider, so every entry must be filled out as "accept" or "reject".

H = Decider for A_{TM}

D = on input $\langle N \rangle$, for TM N

1. Run H on $\langle N, \langle N \rangle \rangle$.
2. If H accepts, reject.
   If H rejects, accept.

Computability Hierarchy



Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega \rangle : M$ is a TM and M halts on ω} is undecidable.

Proof:

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega\rangle : M$ is a TM and M halts on ω} is undecidable.

Proof:

We are going to show that a decider for $HALT_{TM}$ can be used to build a decider for A_{TM}.

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

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega\rangle : M$ is a TM and M halts on ω} is undecidable.

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

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

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega \rangle : M$ is a TM and M halts on ω} is undecidable.

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

Build a TM S that decides A_{TM}:

?

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

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega \rangle : M$ is a TM and M halts on ω} is undecidable.

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

Build a TM S that decides A_{TM}:

S = on input $\langle M, \omega \rangle$

1.

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

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega \rangle : M$ is a TM and M halts on ω} is undecidable.

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

Build a TM S that decides A_{TM}:

S = on input $\langle M, \omega \rangle$

1. Run H on $\langle M, \omega \rangle$.

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

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega\rangle : M$ is a TM and M halts on ω} is undecidable.

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

Build a TM S that decides A_{TM}:

S = on input $\langle M, \omega\rangle$

1. Run H on $\langle M, \omega\rangle$.
2. If H rejects, <u>reject</u> (i.e. M does not halt on ω).

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

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega \rangle : M$ is a TM and M halts on ω} is undecidable.

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

Build a TM S that decides A_{TM}:

S = on input $\langle M, \omega \rangle$

1. Run H on $\langle M, \omega \rangle$.
2. If H rejects, <u>reject</u> (i.e. M does not halt on ω).
3. If H accepts, run M on ω until it halts.

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

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega \rangle : M$ is a TM and M halts on ω} is undecidable.

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

Build a TM S that decides A_{TM}:

S = on input $\langle M, \omega \rangle$

1. Run H on $\langle M, \omega \rangle$.
2. If H rejects, <u>reject</u> (i.e. M does not halt on ω).
3. If H accepts, run M on ω until it halts.
4. If M accepts, <u>accept</u>. If M rejects, <u>reject</u>.

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

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega\rangle : M$ is a TM and M halts on ω} is undecidable.

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

Build a TM S that decides A_{TM}:

S = on input $\langle M, \omega\rangle$

1. Run H on $\langle M, \omega\rangle$.
2. If H rejects, <u>reject</u> (i.e. M does not halt on ω).
3. If H accepts, run M on ω until it halts.
4. If M accepts, <u>accept</u>. If M rejects, <u>reject</u>.

S is a decider for A_{TM}, which is a contradiction.

$\therefore HALT_{TM}$ is undecidable.

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega \rangle : M$ is a TM and M halts on $\omega\}$ is undecidable.

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

Build a TM S that decides A_{TM}:

S = on input $\langle M, \omega \rangle$

1. Run H on $\langle M, \omega \rangle$.
2. If H rejects, <u>reject</u> (i.e. M does not halt on ω).
3. If H accepts, run M on ω until it halts.
4. If M accepts, <u>accept</u>. If M rejects, <u>reject</u>.

S is a decider for A_{TM}, which is a contradiction.

$\therefore$ $HALT_{TM}$ is undecidable.

There is sort of a undecidability proof "blueprint", but it is not as helpful.

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega \rangle : M$ is a TM and M halts on ω} is undecidable.

New problem

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

Build a TM S that decides A_{TM}:

S = on input $\langle M, \omega \rangle$

1. Run H on $\langle M, \omega \rangle$.
2. If H rejects, reject (i.e. M does not halt on ω).
3. If H accepts, run M on ω until it halts.
4. If M accepts, accept. If M rejects, reject.

S is a decider for A_{TM}, which is a contradiction.

$\therefore$ $HALT_{TM}$ is undecidable.

There is sort of a undecidability proof "blueprint", but it is not as helpful.

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega\rangle : M$ is a TM and M halts on ω} is undecidable.

New problem

Proof: **Suppose $HALT_{TM}$ is decidable and let TM $\boldsymbol{H}$ be its decider.**

Known undecidable problem

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

$\boldsymbol{S}$ **= on input** $\langle M, \omega\rangle$

1. Run H on $\langle M, \omega\rangle$.
2. If H rejects, reject (i.e. M does not halt on ω).
3. If H accepts, run M on ω until it halts.
4. If M accepts, accept. If M rejects, reject.

$\boldsymbol{S}$ is a decider for A_{TM}, which is a contradiction.

$\therefore$ $HALT_{TM}$ is undecidable.

There is sort of a undecidability proof "blueprint", but it is not as helpful.

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega \rangle : M$ is a TM and M halts on ω} is undecidable.

New problem

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

Known undecidable problem

Build a TM S that decides A_{TM}:

S = on input $\langle M, \omega \rangle$

Input to known problem

1. Run H on $\langle M, \omega \rangle$.
2. If H rejects, reject (i.e. M does not halt on ω).
3. If H accepts, run M on ω until it halts.
4. If M accepts, accept. If M rejects, reject.

S is a decider for A_{TM}, which is a contradiction.

∴ $HALT_{TM}$ is undecidable.

There is sort of a undecidability proof "blueprint", but it is not as helpful.

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega \rangle : M$ is a TM and M halts on ω$\}$ is undecidable.

New problem

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

Known undecidable problem

Build a TM S that decides A_{TM}:

S = on input $\langle M, \omega \rangle$

Input to known problem

1. Run H on $\langle M, \omega \rangle$.
2. If H rejects, reject (i.e. M does not halt on ω).
3. If H accepts, run M on ω until it halts.
4. If M accepts, accept. If M rejects, reject.

Decider for known problem

S is a decider for A_{TM}, which is a contradiction.

$\therefore HALT_{TM}$ is undecidable.

There is sort of a undecidability proof "blueprint", but it is not as helpful.

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega \rangle: M$ is a TM and M halts on ω} is undecidable.

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

Build a TM S that decides A_{TM}:

S = on input $\langle M, \omega \rangle$

1. Run H on $\langle M, \omega \rangle$.
2. If H rejects, <u>reject</u> (i.e. M does not halt on ω).
3. If H accepts, run M on ω until it halts.
4. If M accepts, <u>accept</u>. If M rejects, <u>reject</u>.

S is a decider for A_{TM}, which is a contradiction.

$\therefore HALT_{TM}$ is undecidable.

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega\rangle : M$ is a TM and M halts on $\omega\}$ is undecidable.

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

Build a TM S that decides A_{TM}:

Used $HALT_{TM}$ decider to make decider for A_{TM}
DID NOT use A_{TM} decider to make decider for $HALT_{TM}$

3. If H accepts, run M on ω until it halts.
4. If M accepts, accept. If M rejects, reject.

S is a decider for A_{TM}, which is a contradiction.

$\therefore HALT_{TM}$ is undecidable.

Undecidability “Direction”

Fact: **Magic Wands Cannot Exist**

Undecidability "Direction"

Fact: Magic Wands Cannot Exist

Use?	Hypothetical SuperSaw 2.0
Do?	Build a magic wand
Conclude?	**?**

Undecidability “Direction”

Fact: Magic Wands Cannot Exist

Use?	Hypothetical SuperSaw 2.0
Do?	Build a magic wand
Conclude?	SuperSaw 2.0 saw cannot exist

Undecidability “Direction”

Fact: **Magic Wands Cannot Exist**

Use?	**Something Assumed to Exist** Hypothetical SuperSaw 2.0
Do?	**Something Impossible** Build a magic wand
Conclude?	**Assumed Thing Cannot Exist** SuperSaw 2.0 saw cannot exist

Undecidability "Direction"

Fact: **Magic Wands Cannot Exist**

Use?	**Something Assumed to Exist** $HALT_{TM}$ Decider
Do?	**Something Impossible** Build an A_{TM} Decider
Conclude?	**Assumed Thing Cannot Exist** **?**

Undecidability "Direction"

Fact: **Magic Wands Cannot Exist**

Use?	**Something Assumed to Exist** $HALT_{TM}$ Decider
Do?	**Something Impossible** Build an A_{TM} Decider
Conclude?	**Assumed Thing Cannot Exist** $HALT_{TM}$ Decider can't exist

Undecidability "Direction"

Fact: **Magic Wands Cannot Exist**

Use?	Magic Wand	**Something Assumed to Exist** $HALT_{TM}$ Decider
Do?	Bake a Cake	**Something Impossible** Build an A_{TM} Decider
Conclude?	**?**	**Assumed Thing Cannot Exist** $HALT_{TM}$ Decider can't exist

Undecidability "Direction"

Fact: **Magic Wands Cannot Exist**

Use?	Magic Wand	**Something Assumed to Exist** $HALT_{TM}$ Decider
Do?	Bake a Cake	**Something Impossible** Build an A_{TM} Decider
Conclude?	**Nothing**	**Assumed Thing Cannot Exist** $HALT_{TM}$ Decider can't exist

Undecidability "Direction"

Fact: **Magic Wands Cannot Exist**

Use?	Magic Wand	**Something Assumed to Exist** $HALT_{TM}$ Decider
Do?	Go Back in Time	**Something Impossible** Build an A_{TM} Decider
Conclude?		**Assumed Thing Cannot Exist** $HALT_{TM}$ Decider can't exist

Undecidability "Direction"

Fact: Magic Wands Cannot Exist

Use?	Magic Wand	**Something Assumed to Exist** $HALT_{TM}$ Decider
Do?	Go Back in Time	**Something Impossible** Build an A_{TM} Decider
Conclude?	**Nothing**	**Assumed Thing Cannot Exist** $HALT_{TM}$ Decider can't exist

Undecidability "Direction"

Fact: **Magic Wands Cannot Exist**

Use?	**Something Impossible** Magic Wand	**Something Assumed to Exist** $HALT_{TM}$ Decider
Do?	**Any Task** Go Back in Time	**Something Impossible** Build an A_{TM} Decider
Conclude?		**Assumed Thing Cannot Exist** $HALT_{TM}$ Decider can't exist

Undecidability "Direction"

Fact: **Magic Wands Cannot Exist**

Use?	**Something Impossible** Magic Wand	**Something Assumed to Exist** $HALT_{TM}$ Decider
Do?	**Any Task** Go Back in Time	**Something Impossible** Build an A_{TM} Decider
Conclude?	**Nothing**	**Assumed Thing Cannot Exist** $HALT_{TM}$ Decider can't exist

Undecidability "Direction"

<u>Fact:</u> Magic Wands Cannot Exist

Use?	**Something Impossible** A_{TM} Decider	**Something Assumed to Exist** $HALT_{TM}$ Decider
Do?	**Any Task** Build $HALT_{TM}$ Decider	**Something Impossible** Build an A_{TM} Decider
Conclude?		**Assumed Thing Cannot Exist** $HALT_{TM}$ Decider can't exist

Undecidability "Direction"

Fact: **Magic Wands Cannot Exist**

Use?	**Something Impossible** A_{TM} Decider	**Something Assumed to Exist** $HALT_{TM}$ Decider
Do?	**Any Task** Build $HALT_{TM}$ Decider	**Something Impossible** Build an A_{TM} Decider
Conclude?	**Nothing**	**Assumed Thing Cannot Exist** $HALT_{TM}$ Decider can't exist

Halting Problem

Claim: $HALT_{TM} = \{\langle M, \omega\rangle : M$ is a TM and M halts on ω} is undecidable.

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

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

S = on input $\langle M, \omega\rangle$

1. Run H on $\langle M, \omega\rangle$.
2. If H rejects, <u>reject</u> (i.e. M does not halt on ω).
3. If H accepts, run M on ω until it halts.
4. If M accepts, <u>accept</u>. If M rejects, <u>reject</u>.

S is a decider for A_{TM}, which is a contradiction.

∴ **$HALT_{TM}$ is undecidable**.