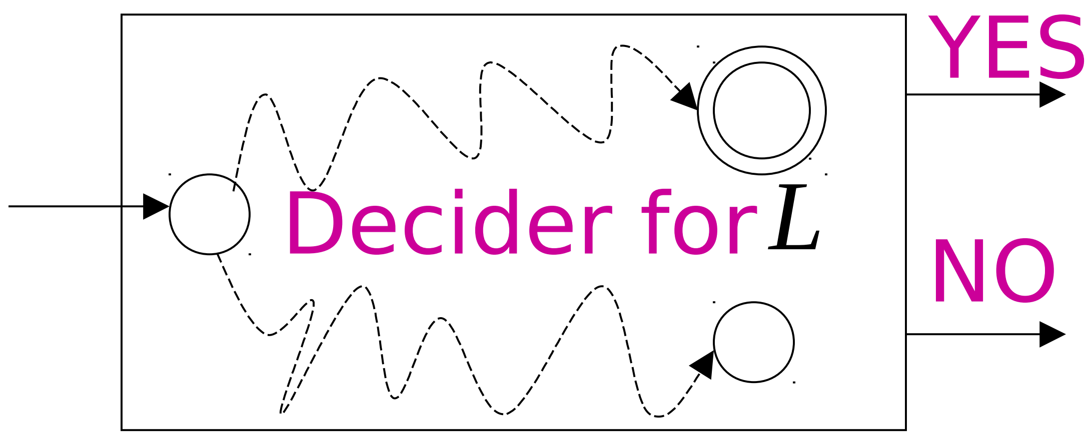
Undecidable Problems

Recall that:

A language L is decidable,

if there is a Turing machine M (decider)

that accepts L and halts on every input stri



Undecidable Language L

There is no decider for L :

there is no Turing Machine
which accepts L
and halts on every input string

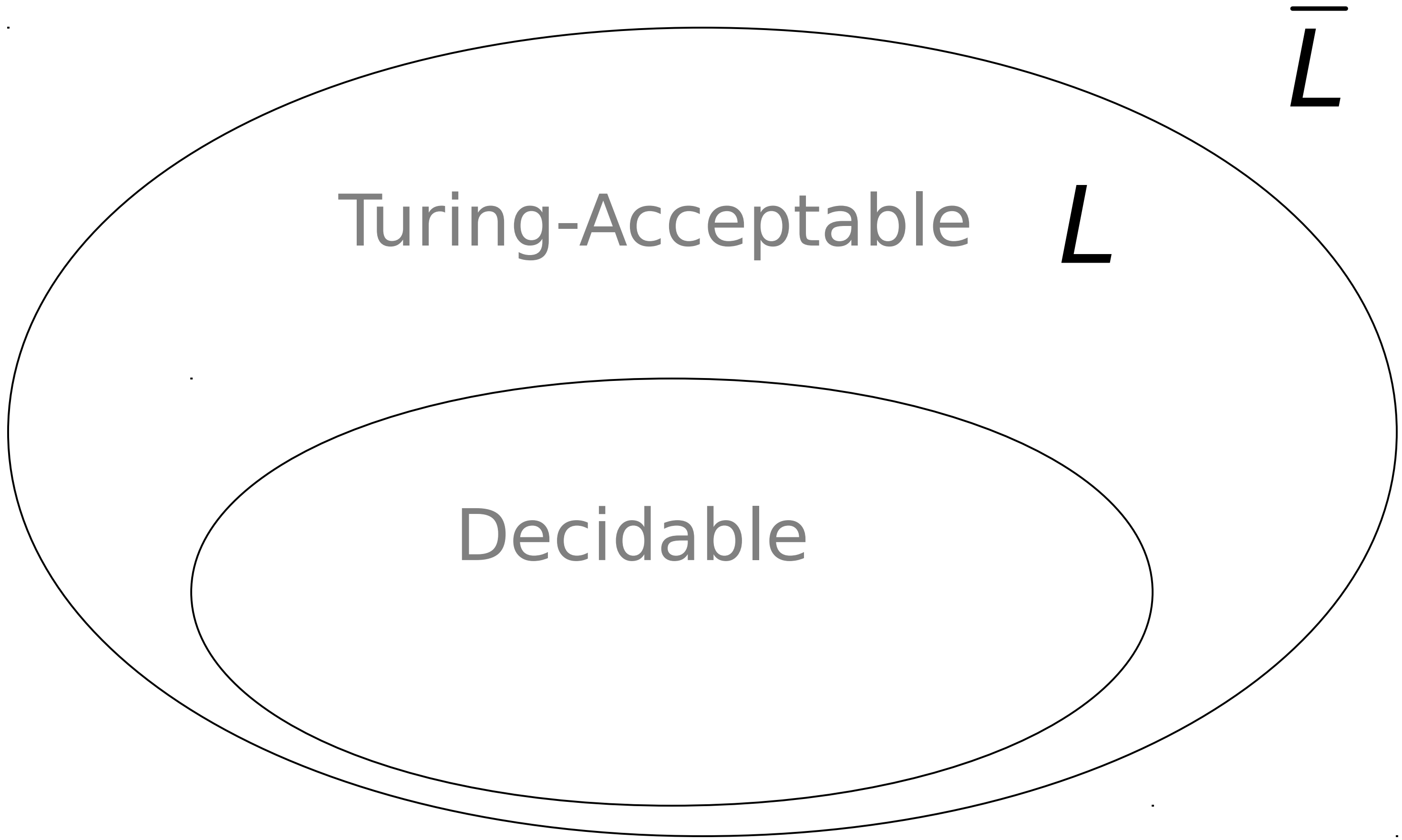
(the machine may halt and decide for some input strings)

For an undecidable language,
the corresponding problem is
undecidable (unsolvable):

there is no Turing Machine (Algorithm)
that gives an answer (yes or no)
for every input instance

(answer may be given for some input instances)

We have shown before that there are undecidable languages:



L is Turing-Acceptable and undecidable

We will prove that two particular problems are unsolvable:

Membership problem

Halting problem

Membership Problem

Input: • Turing Machine M

• String w

Question: Does M accept w ?

$$w \in L(M)?$$

Corresponding language:

$$A_{TM} = \{\langle M, w \rangle : M \text{ is a Turing machine that accepts string } w\}$$

Theorem: A_{TM} is undecidable

(The membership problem is unsolvable)

Proof:

Basic idea:

We will assume that A_{TM} is decidable;

We will then prove that

every Turing-acceptable language

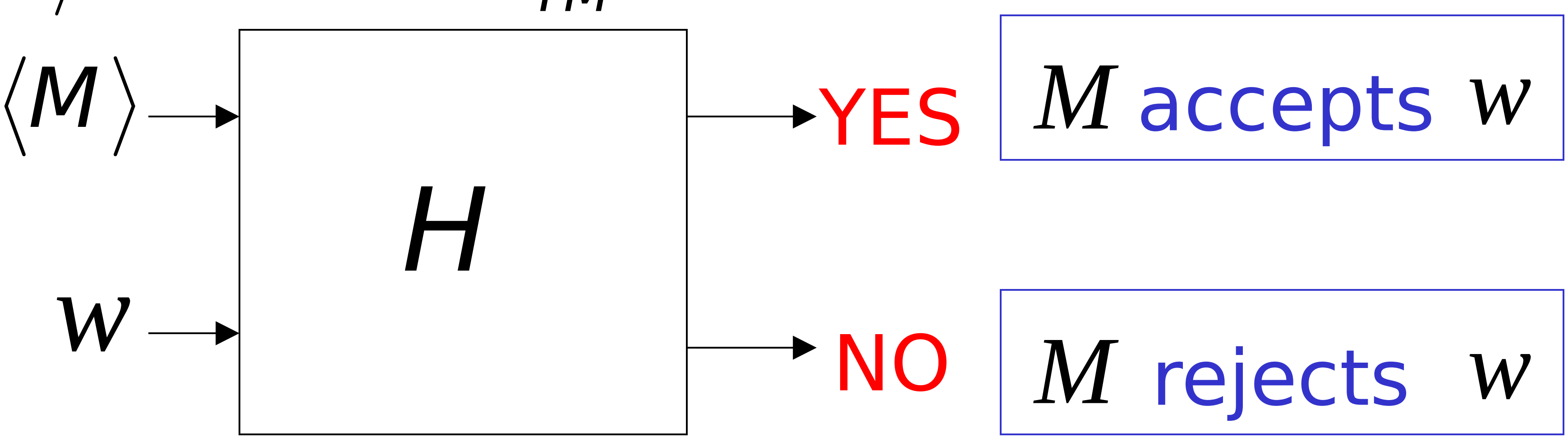
is also decidable

A contradiction!

Suppose that A_{TM} is decidable

Input string $\langle M, w \rangle$

Decider for A_{TM}



et L be a Turing recognizable language

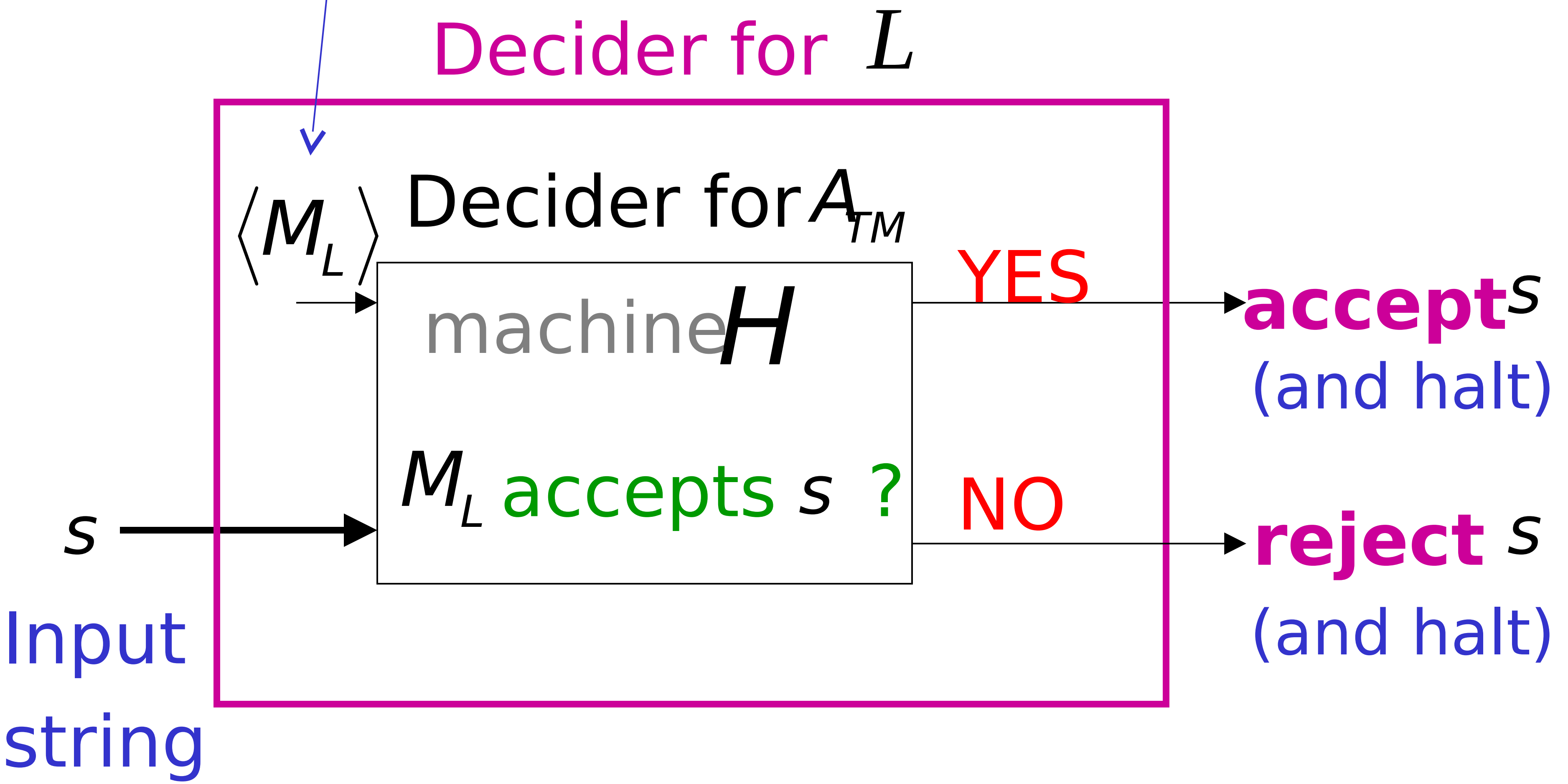
Let M_L be the Turing Machine that accepts L

We will prove that L is also decidable:

we will build a decider for L

String description of M_L

This is hardwired and copied on the tape next to input string s, and then the pair $\langle M_L, s \rangle$ is input to H



Therefore, L is decidable

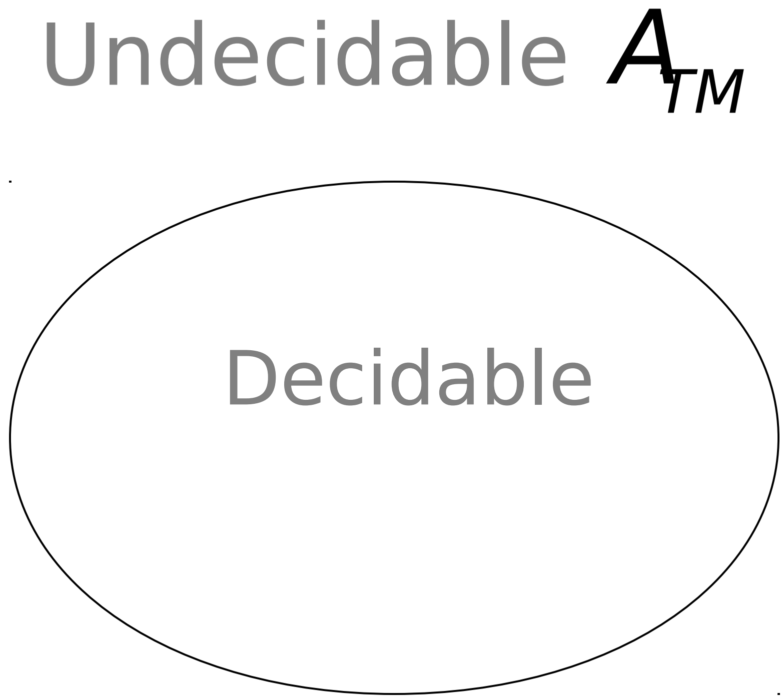
Since L is chosen arbitrarily, every Turing-Acceptable language is decidable

But there is a Turing-Acceptable language which is undecidable

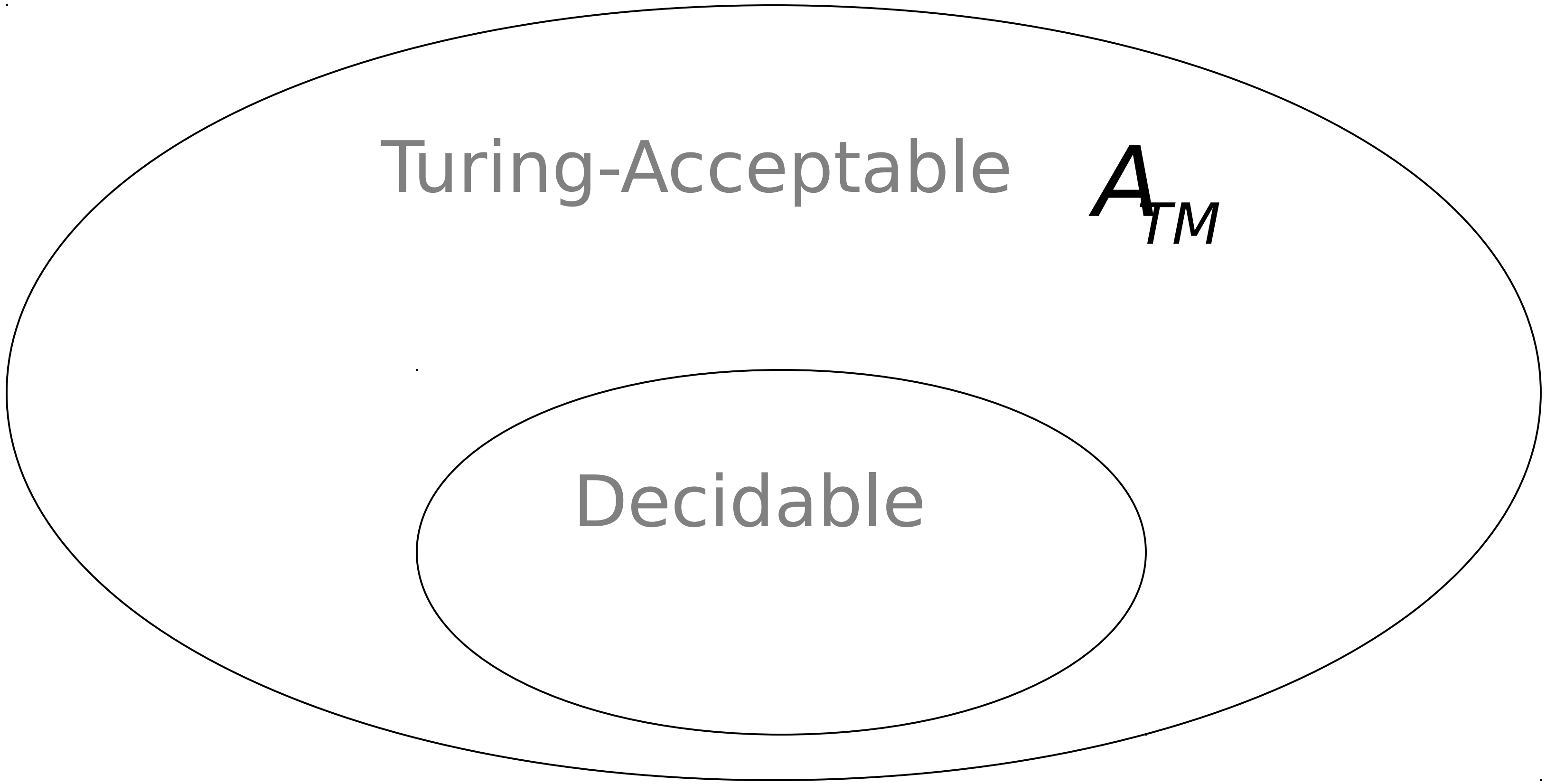
Contradiction!!!!

END OF PROOF

We have shown:



We can actually show:



A_{TM} is Turing-Acceptable

Turing machine that accepts A_{TM} :

$\langle M, w \rangle \rightarrow$

1. Run M on input w

2. If M accepts w
   then accept $\langle M, w \rangle$

Halting Problem

Input: • Turing Machine M
• String w

Question: Does M halt while processing input string w ?

Corresponding language:

$$HALT_{TM} = \{\langle M, w \rangle : M \text{ is a Turing machine that halts on input string } w\}$$

Theorem: $HALT_{TM}$ is undecidable

(The halting problem is unsolvable)

Proof:

Basic idea:

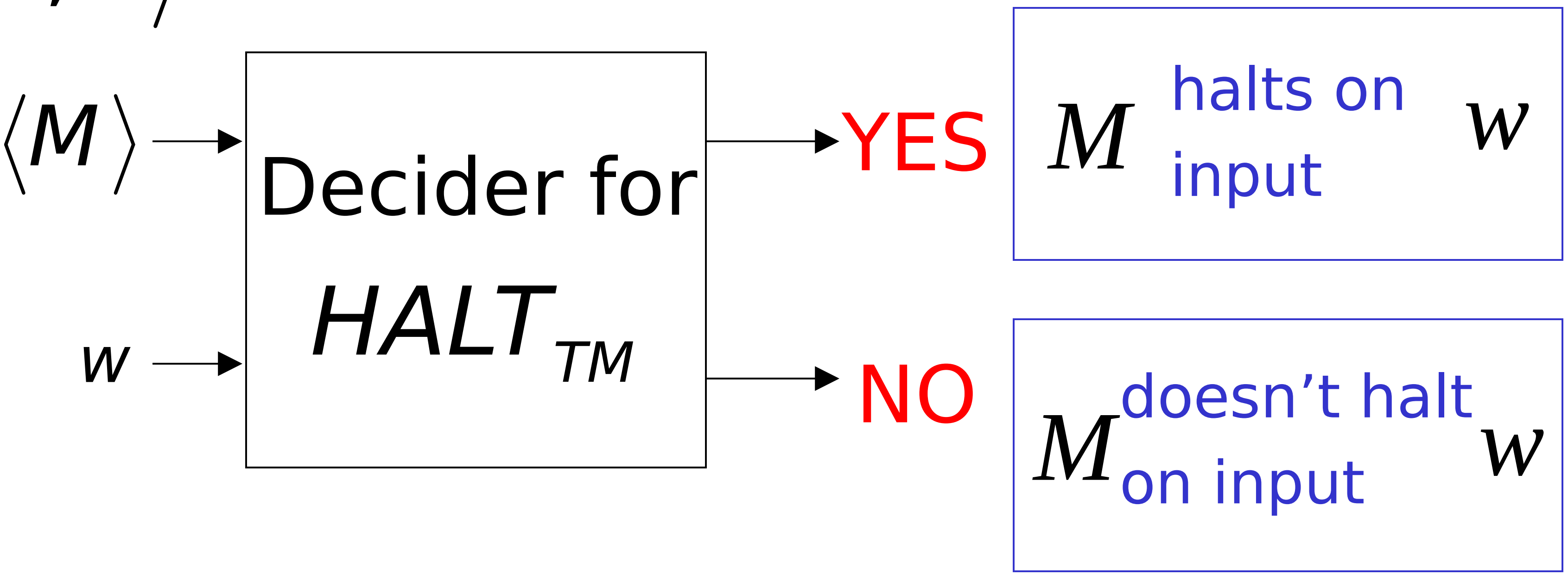
Suppose that $HALT_{TM}$ is decidable;
we will prove that
every Turing-acceptable language
is also decidable

A contradiction!

Suppose that $HALT_{TM}$ is decidable

Input string

$\langle M, w \rangle$



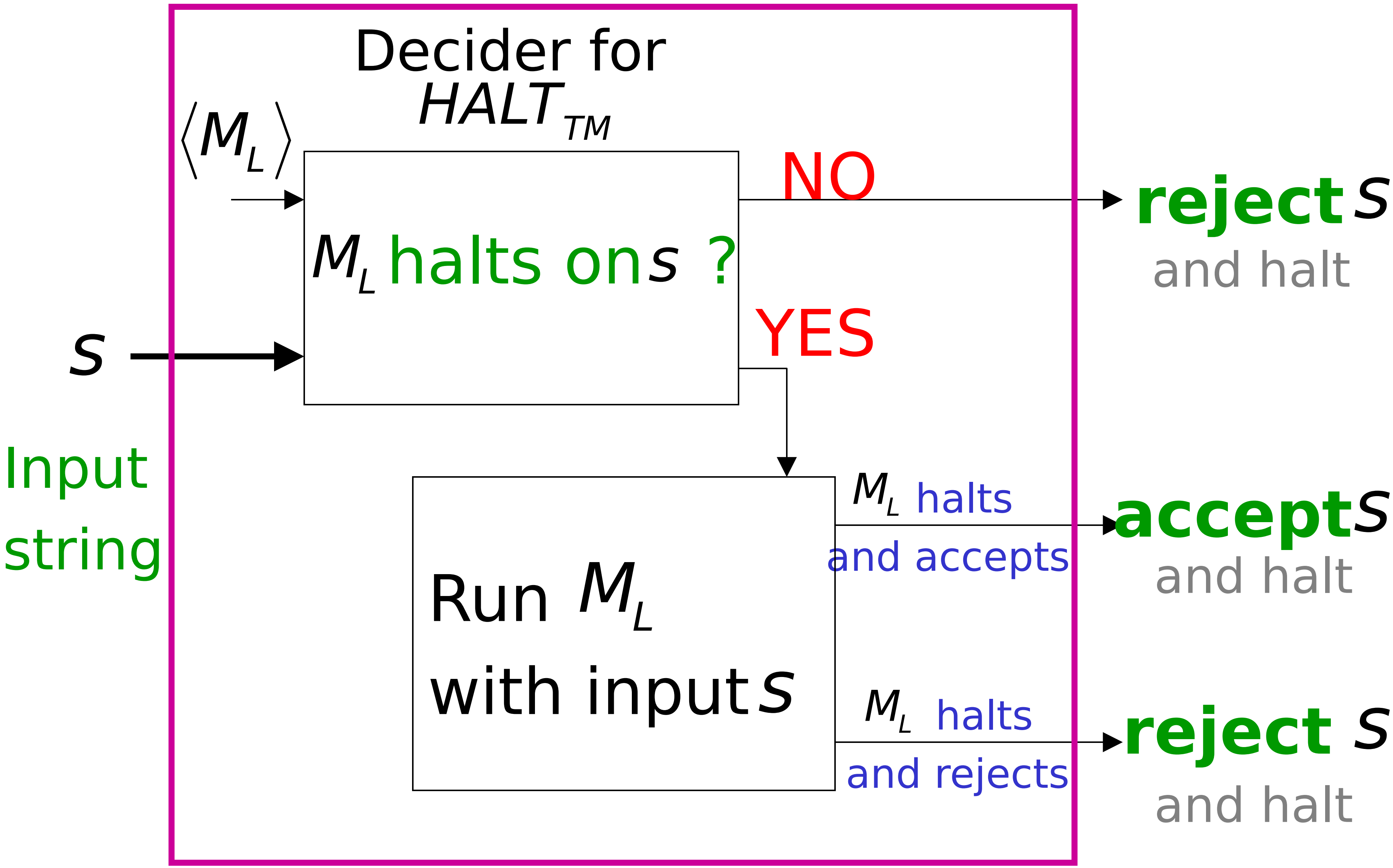
Let L be a Turing-Acceptable language

Let M_L be the Turing Machine that accepts L

We will prove that L is also decidable:

we will build a decider for L

Decider for L



Therefore, L is decidable

Since L is chosen arbitrarily, every Turing-Acceptable language is decidable

But there is a Turing-Acceptable language which is undecidable

Contradiction!!!!

END OF PROOF

An alternative proof

Theorem: $HALT_{TM}$ is undecidable

(The halting problem is unsolvable)

Proof:

Basic idea:

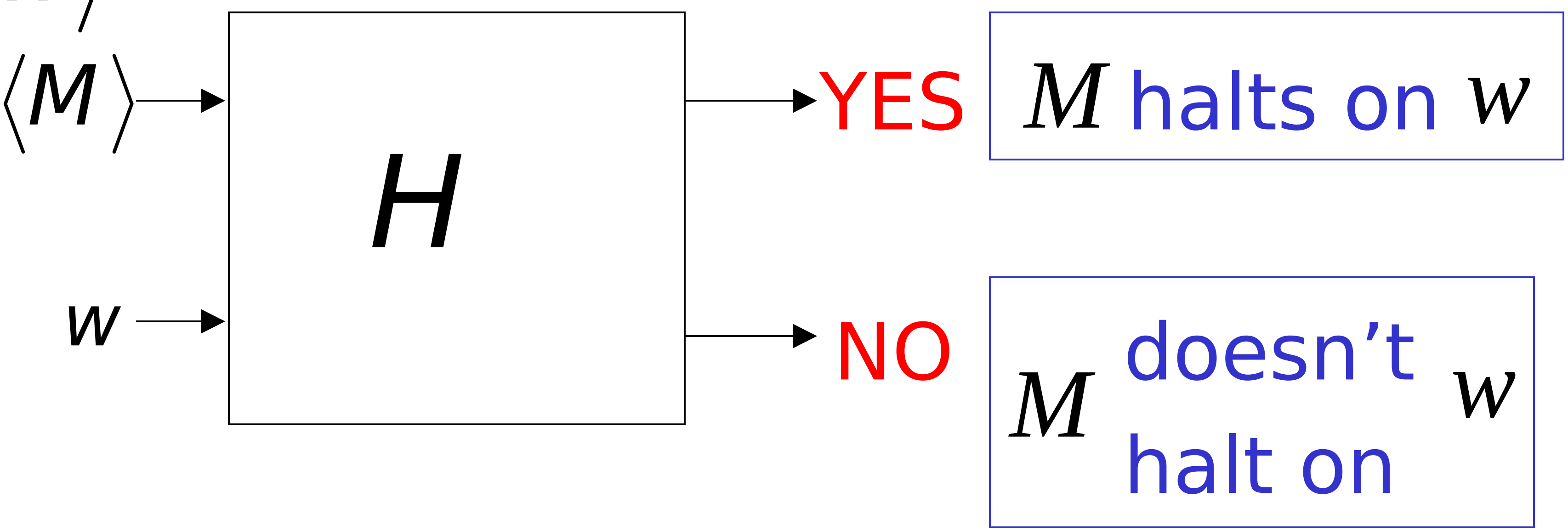
Assume for contradiction that
the halting problem is decidable;

we will obtain a contradiction
using a diagonilization technique

Suppose that $HALT_{TM}$ is decidable

Input string $\langle M, w \rangle$

Decider for $HALT_{TM}$

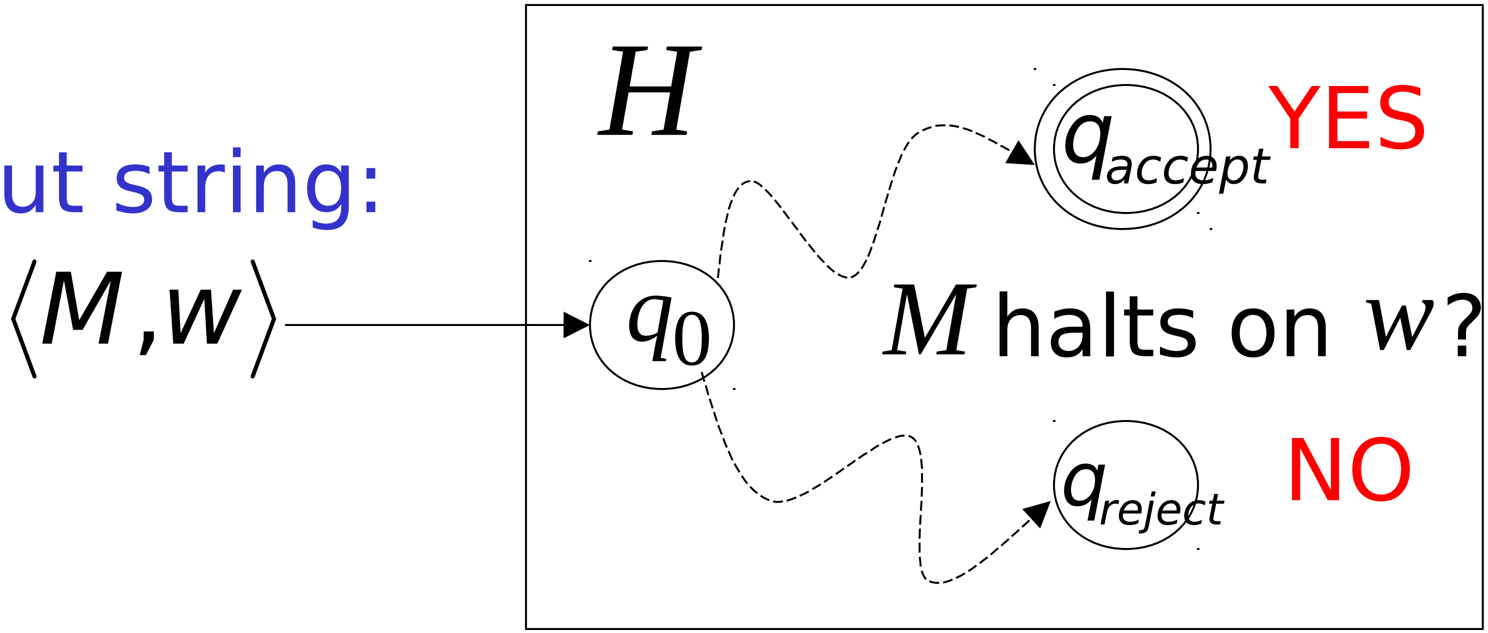


Looking inside H

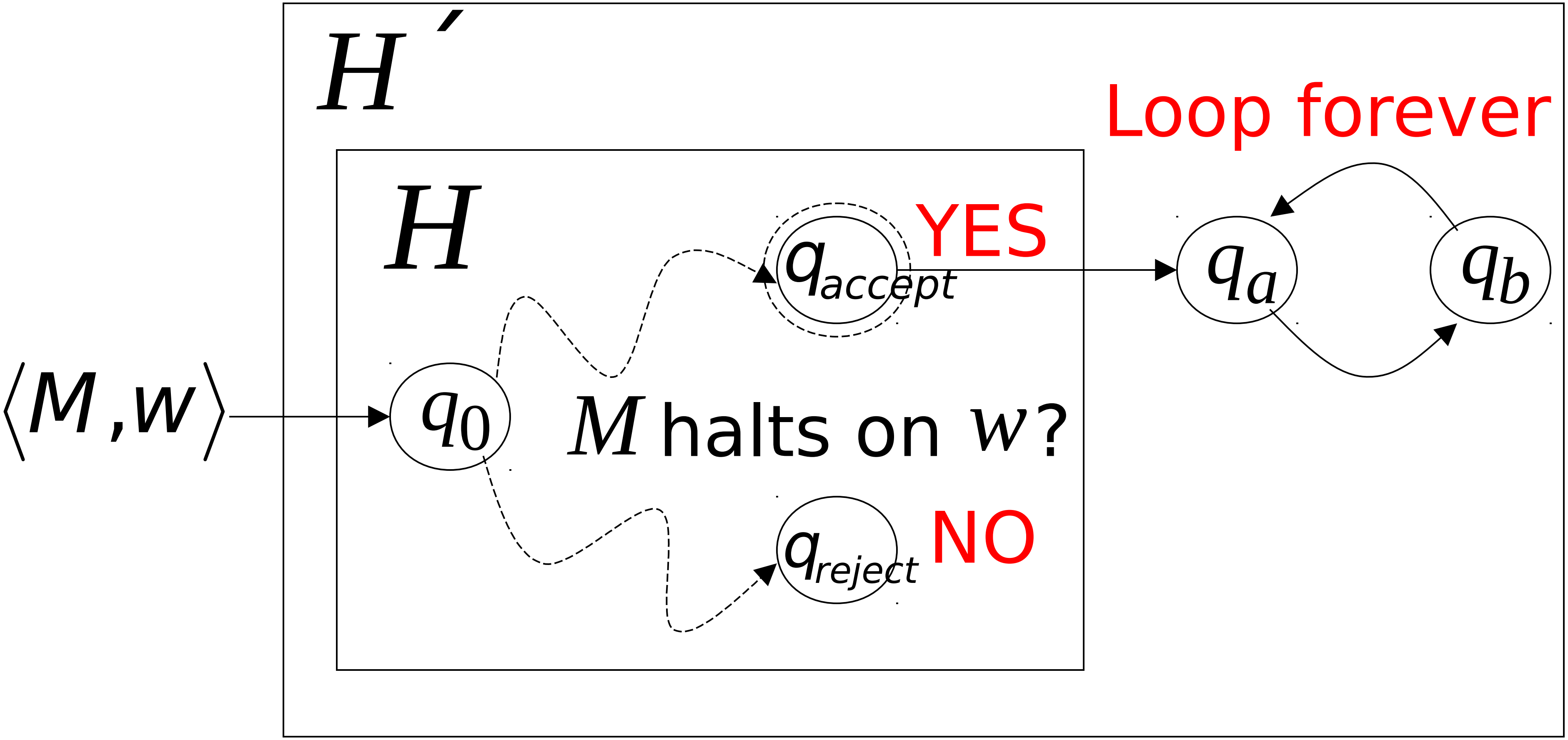
Decider for $HALT_{TM}$

Input string:

$\langle M, w \rangle$



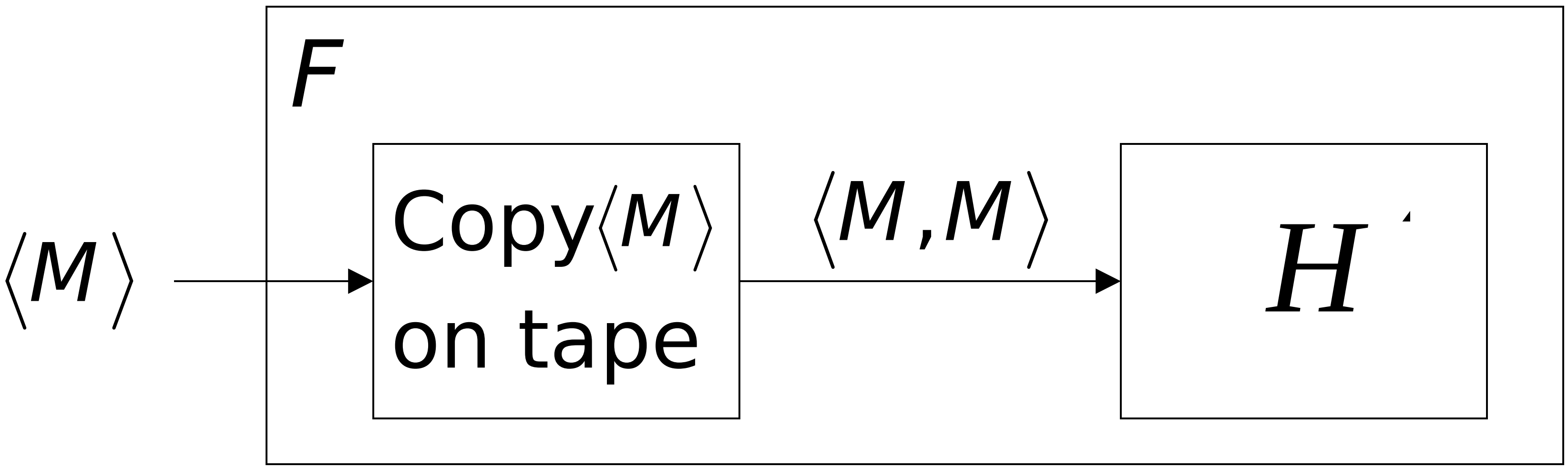
Construct machine H' :



If M halts on input w **<u>Then</u>** Loop Forever

<u>Else</u> Halt

Construct machine F :

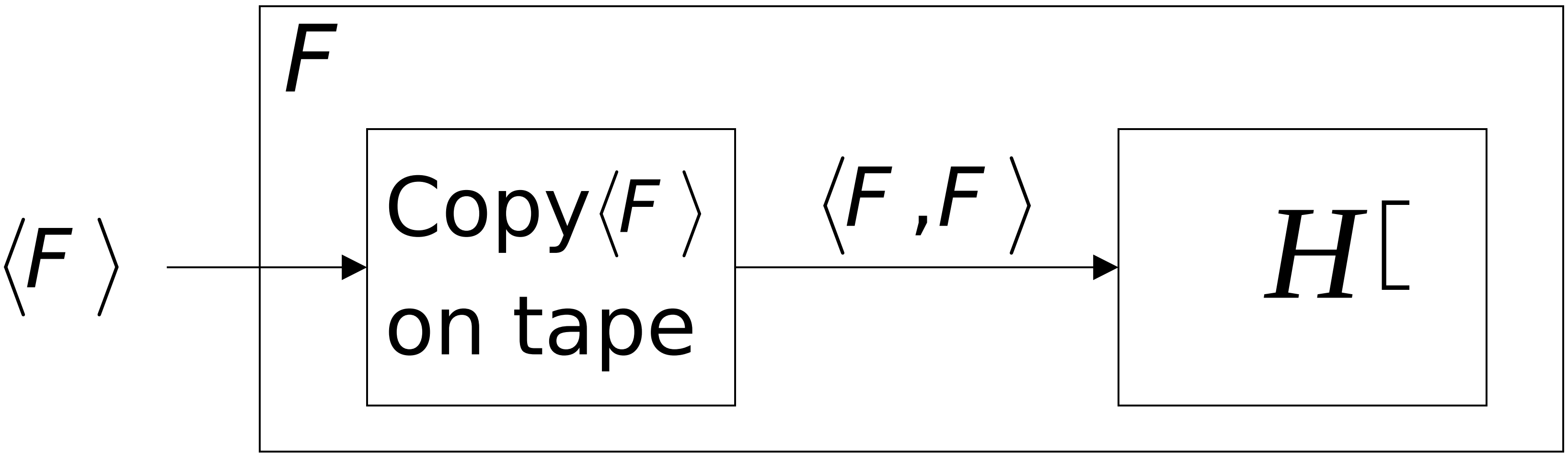


If M halts on input $\langle M \rangle$

Then loop forever

Else halt

Run F with input itself



If F halts on input $\langle F \rangle$

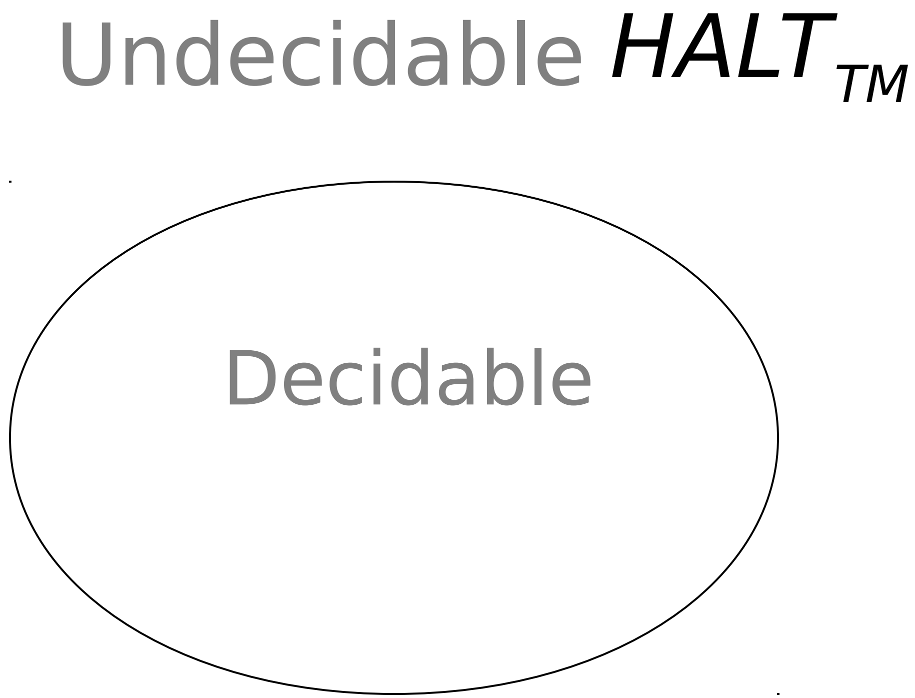
Then F loops forever on input $\langle F \rangle$

Else F halts on input $\langle F \rangle$

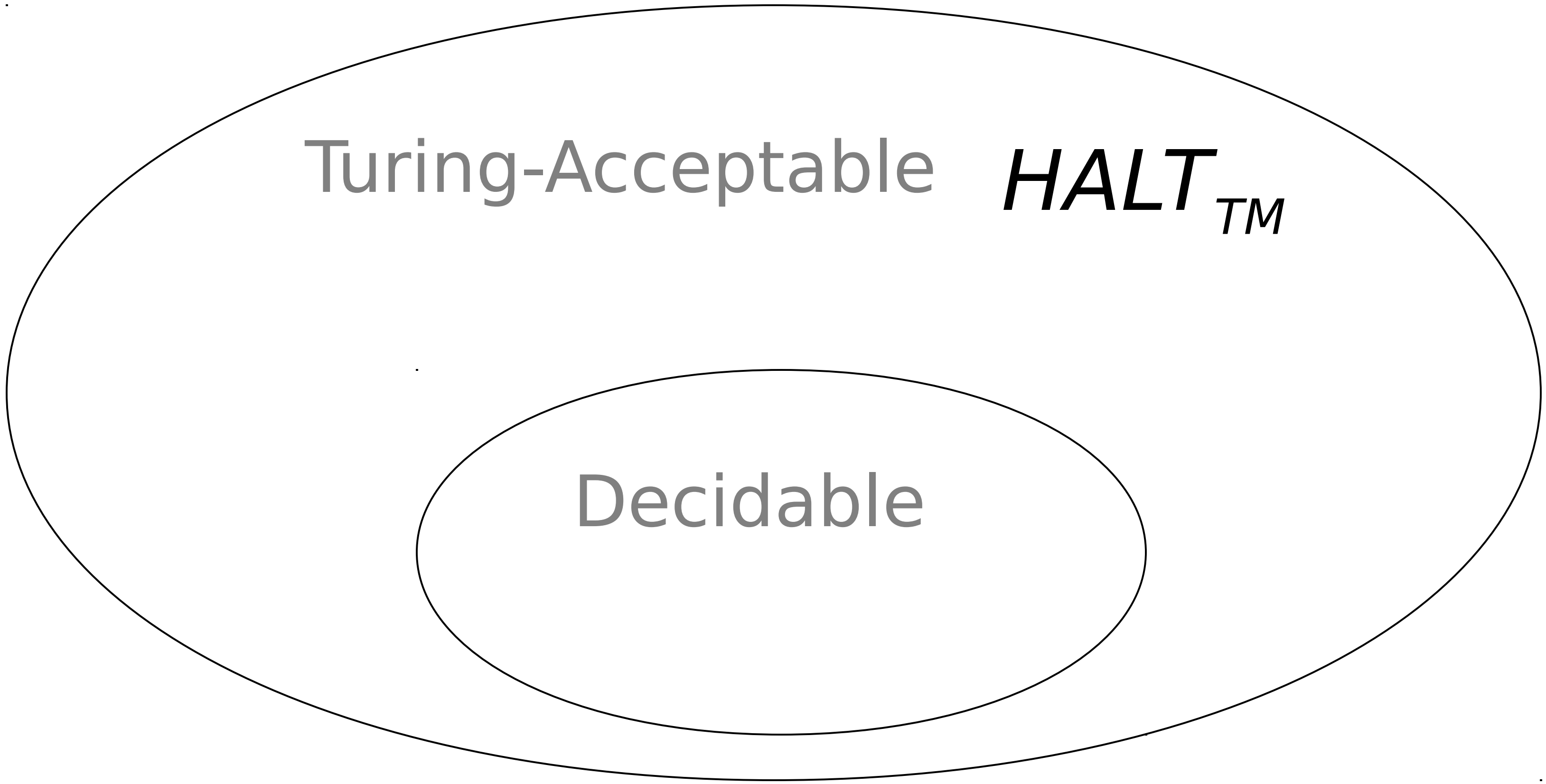
CONTRADICTION!!!

END OF PROOF

We have shown:



We can actually show:



$HALT_{TM}$ is Turing-Acceptable

Turing machine that accepts $HALT_{TM}$:

$\langle M, w \rangle \rightarrow$

1. Run M on input w

2. If M halts on w
   then accept $\langle M, w \rangle$

We showed: