Undecidable Problems
(unsolvable problems)

Undecidable Languages

undecidable language = not decidable language

There is no decider:

there is no Turing Machine
which accepts the language
and makes a decision (halts)
for every input string

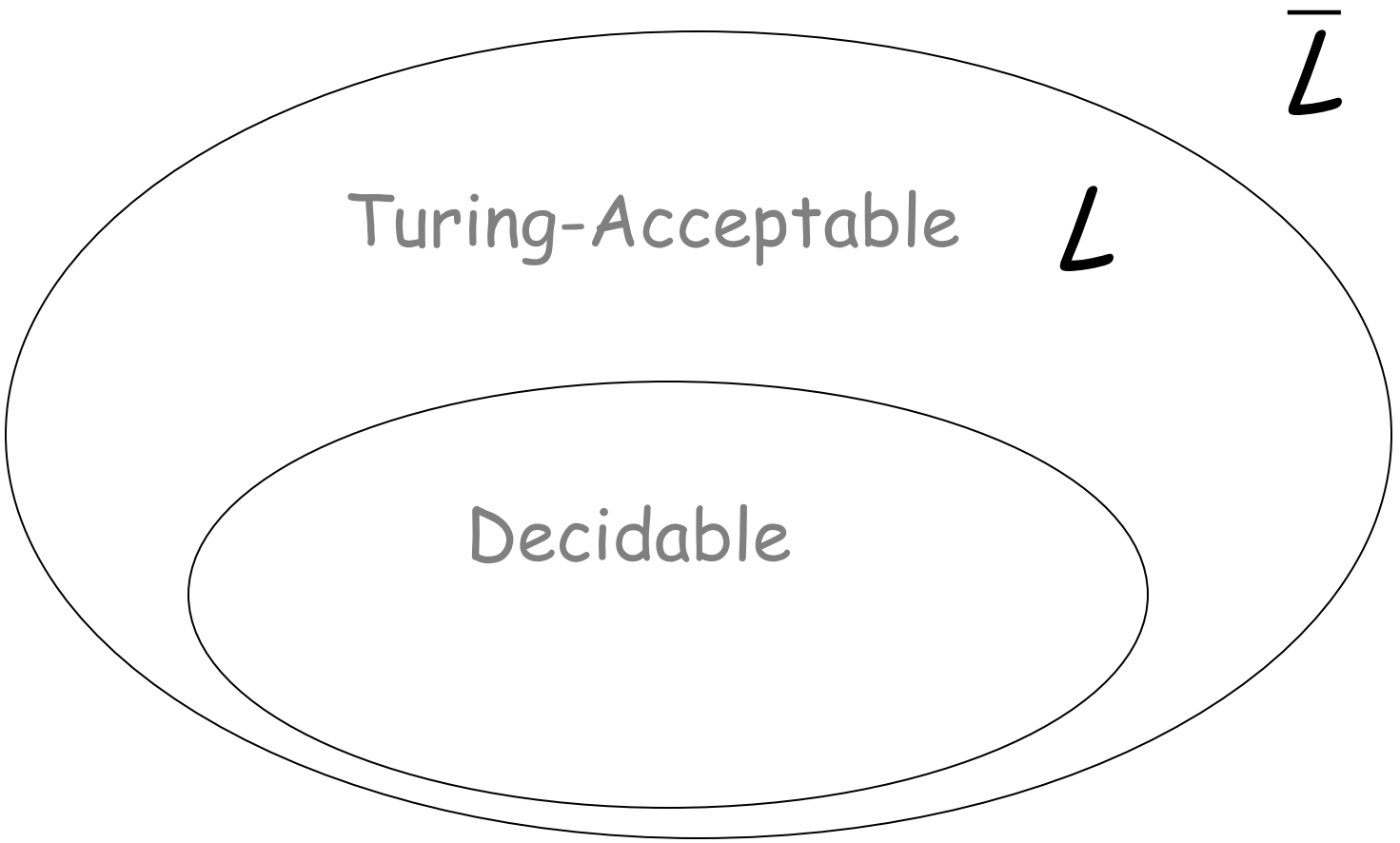
(machine may make decision for some input strings)

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 ?

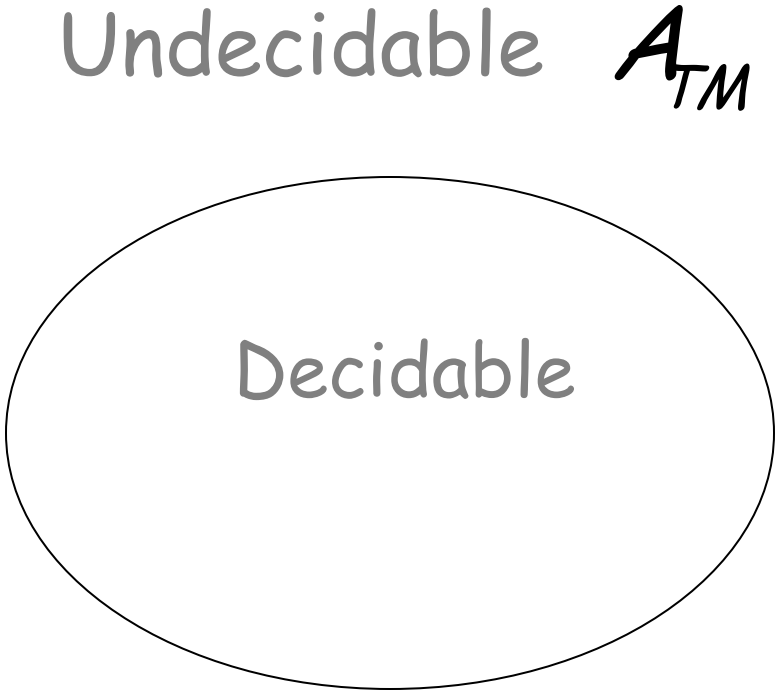
$w \in L(M)$?

Corresponding language:

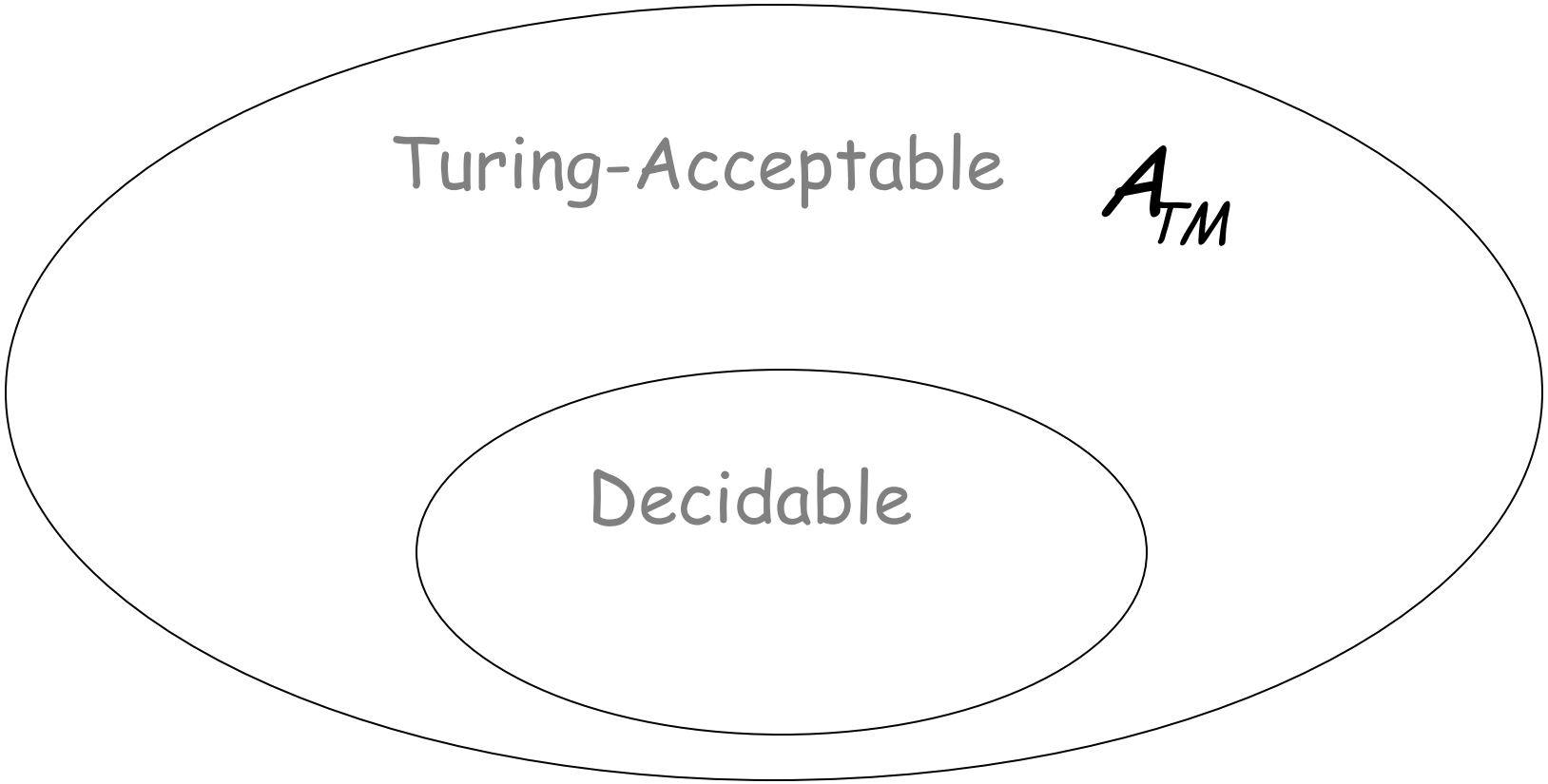
$A_{TM} = \{\langle M, w \rangle : M$ is a Turing machine that accepts string $w\}$

Theorem: A_{TM} is undecidable

(The membership problem is unsolvable)

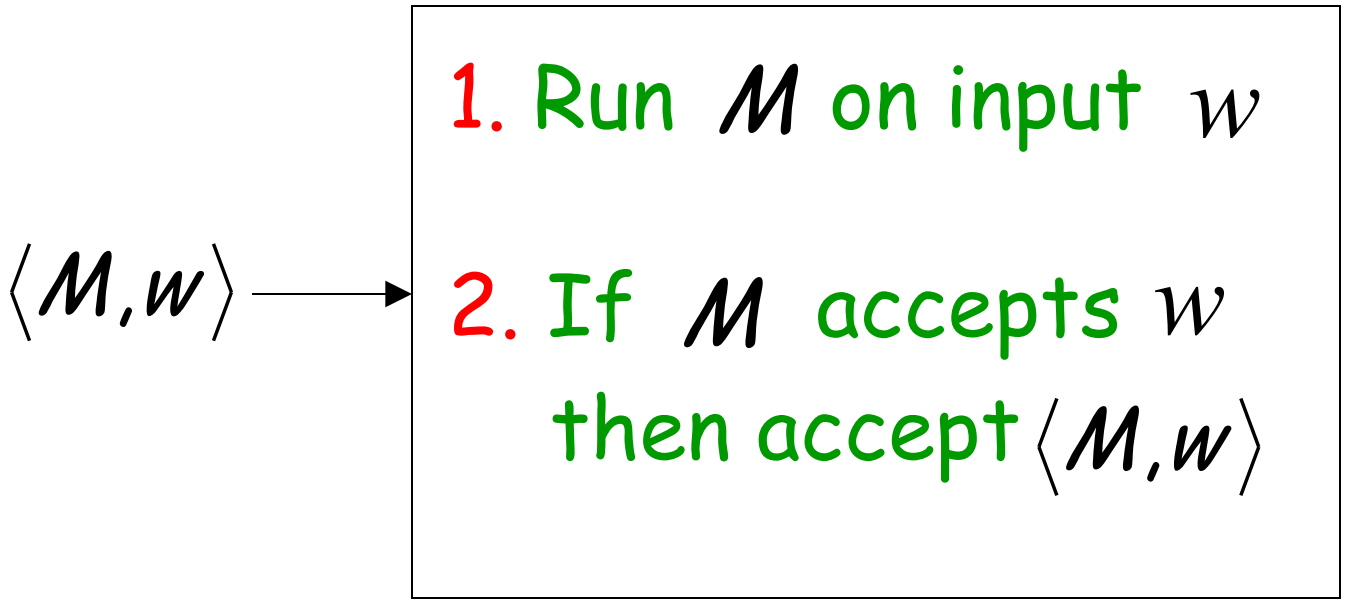


We actually have:



A_{TM} is Turing-Acceptable

Turing machine that accepts A_{TM} :



Halting Problem

Input:
- Turing Machine M
- String w

Question: Does M halt while processing input string w ?

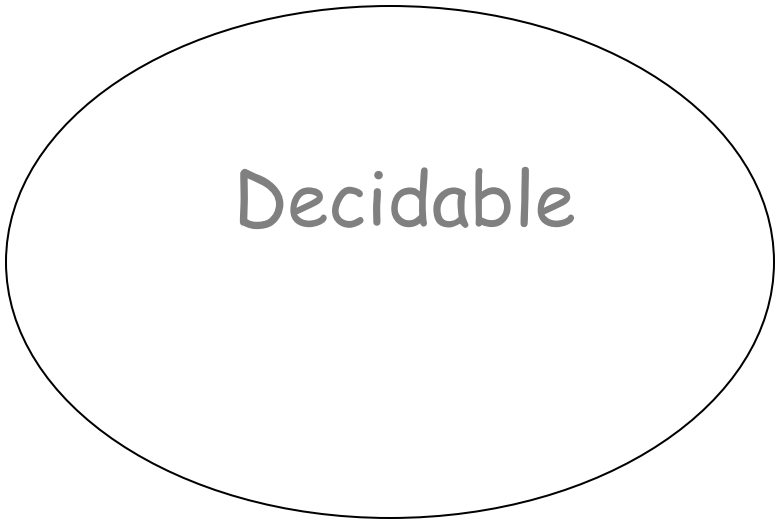
Corresponding language:

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

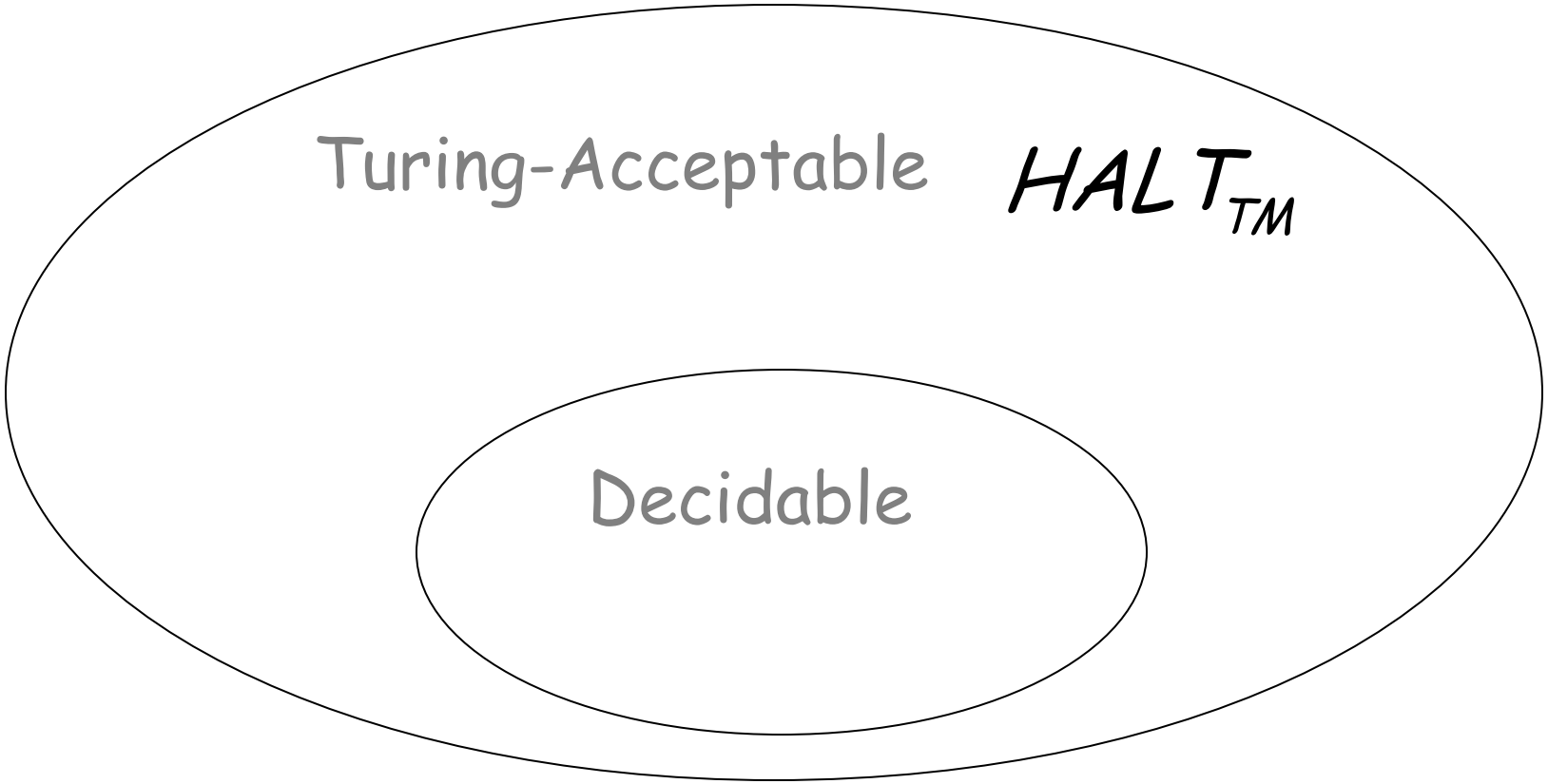
Theorem: $HALT_{TM}$ is undecidable

(The halting problem is unsolvable)

Undecidable $HALT_{TM}$



We actually have:



$HALT_{TM}$ is Turing-Acceptable

Turing machine that accepts $HALT_{TM}$: