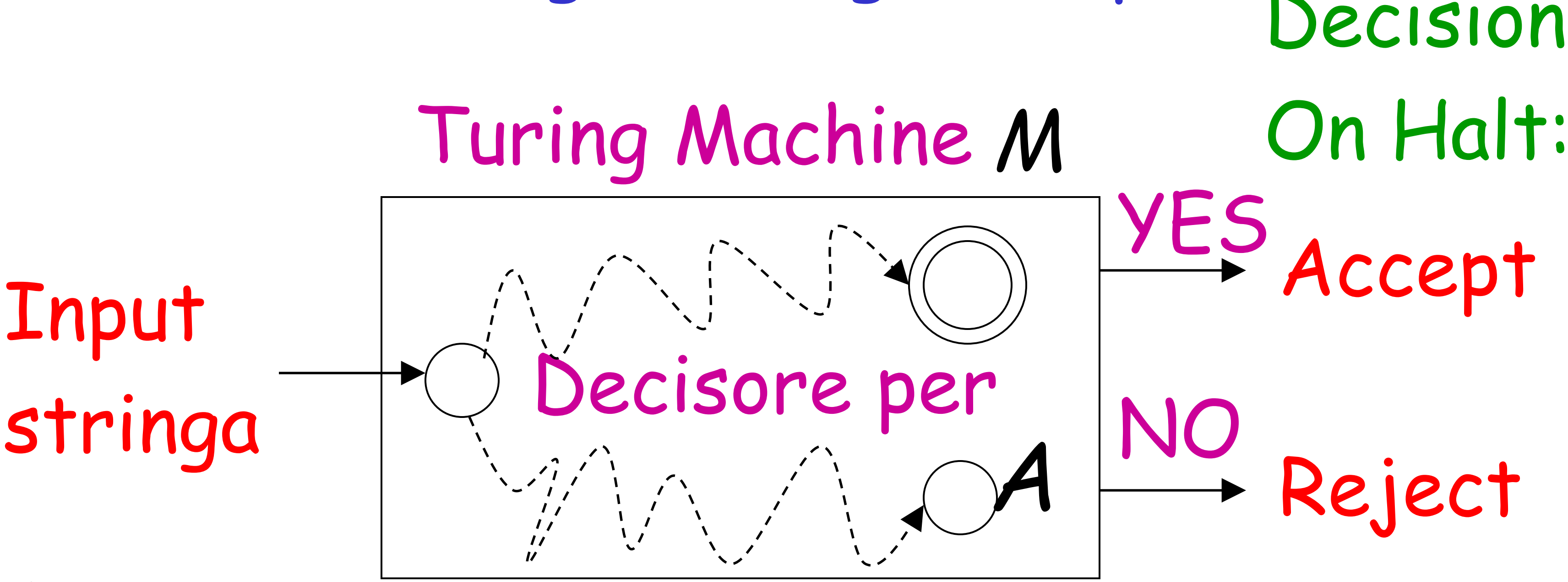
Problemi indecidibili (unsolvable problems)

linguaggi decidibili

Ricordiamo che:

un linguaggio A è decidibile,
se vi è una Turing machine M (decisore)
che accetta il linguaggio A e
e si ferma su ogni stringa di input



Definizione

Un problema computazionale è decidibile se il corrispondente linguaggio è decidibile

linguaggi indecidibili

linguaggi indecidibili = linguaggi non decidibili

Non esiste un procedimento di decisione (decisore):

Non esiste una Turing Machine
che accetta il linguaggio
e <u>prende una decisione</u> (halts)
per ogni stringa di input.

(la macchina può prendere decisioni per qualche stringa ma non per tutte)

Per un linguaggio indecidibile,
Il corrispondente problema è
indecidibile (unsolvable):

Non esiste una Turing Machine (Algorithm)
che per ogni input
dà una risposta
(yes (appartiene al linguaggio)
or no (non appartiene al linguaggio)

(chiaramente per alcuni input si)

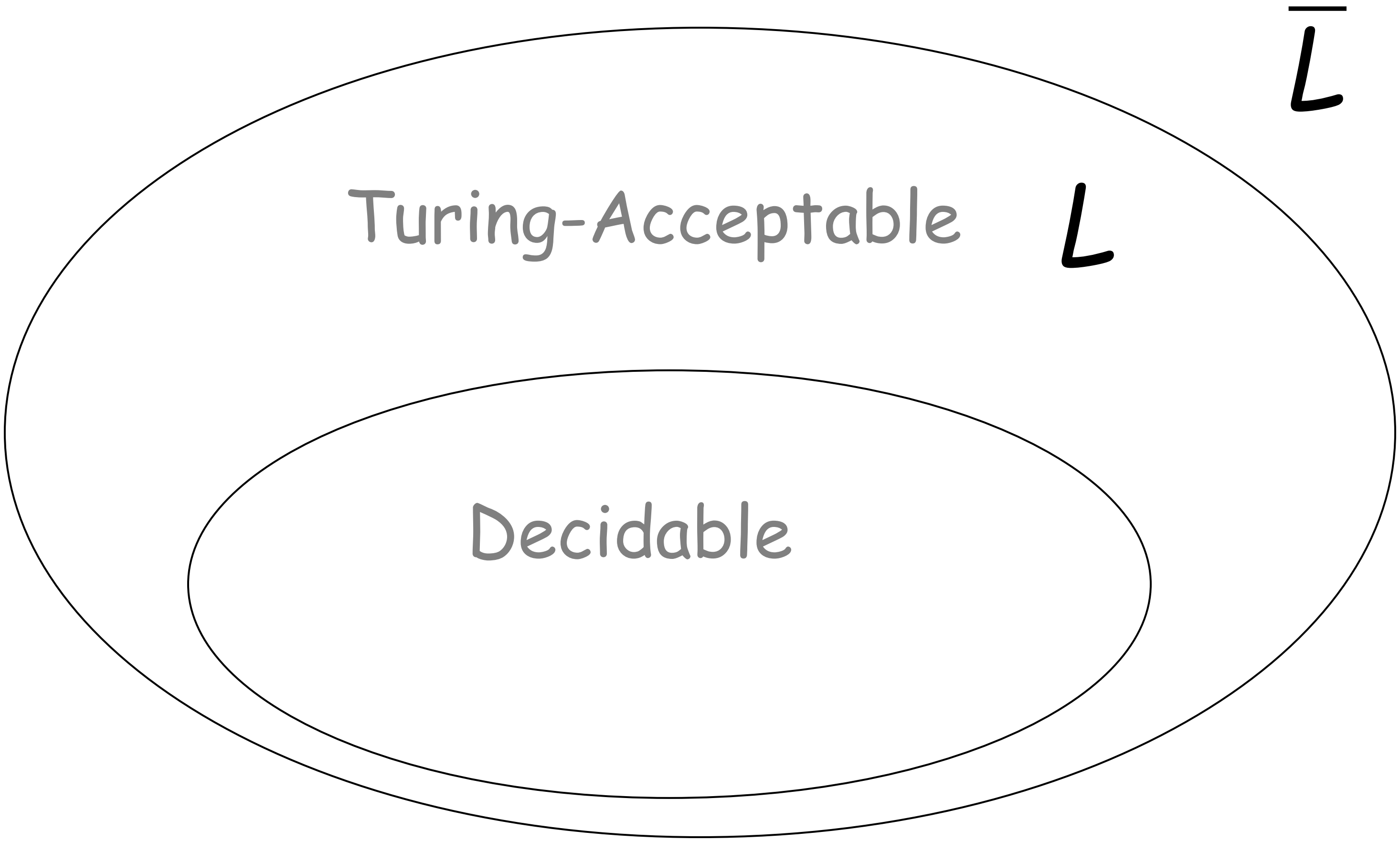
Definizione. Turing accettabile

Abbiamo una Turing Machine (Algorithm) che

1_ per ogni input abbiamo una risposta se la stringa appartiene al linguaggio(yes)

2_ ma nulla possiamo dire se la stringa non appartiene al linguaggio

Abbiamo gia mostrato che esistono linguaggi indecidibili:



Affronteremo due particolari problemi:

Membership problem

Halting problem

Qui 16 5

Membership Problem

Input: • Turing Machine M
• String w

Question: M accetta w?

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

linguaggio corrispondente:

$$A_{TM} = \{\langle M, w\rangle : M \text{ è una Turing machine che accetta la stringa } w\}$$

Teorema: A_{TM} è indecidibile

(The membership problem non è decidible)

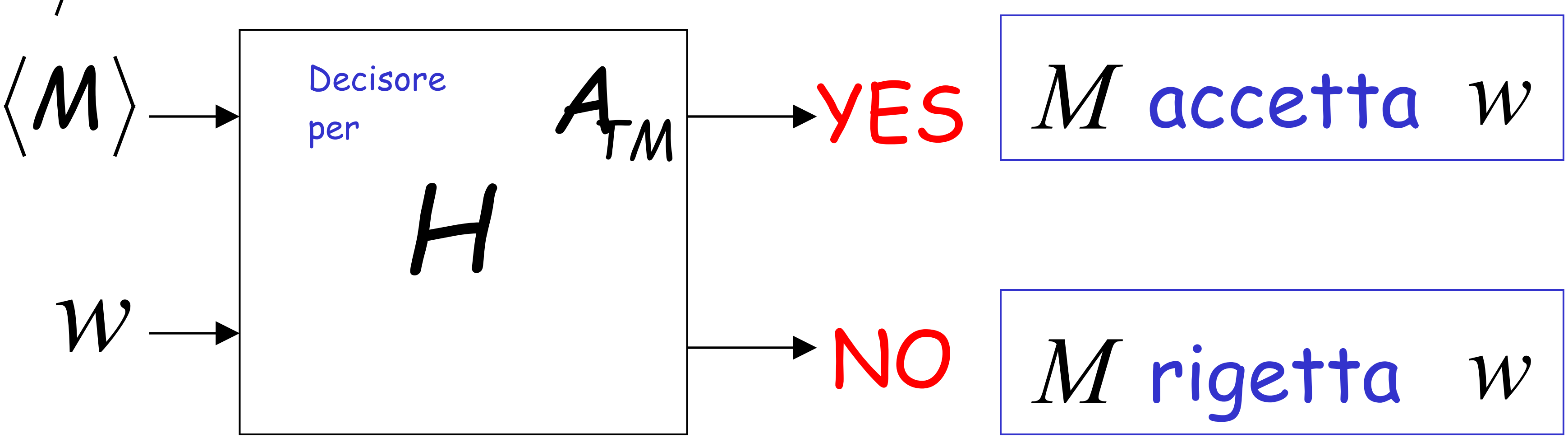
Supponiamo che A_{TM} sia decidibile

Supponiamo che A_{TM} è decidibile

Esiste una macchina H:

Input string

$\langle M, w \rangle$

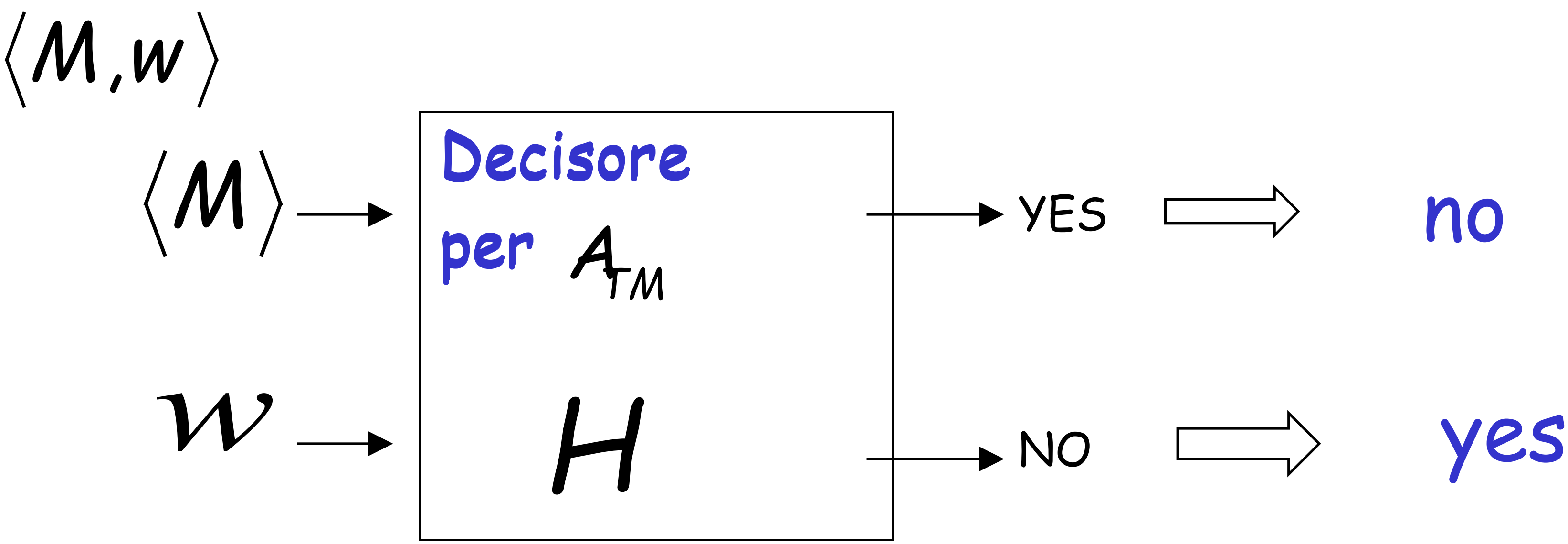


Cambiamo, diagonalizziamo, definiamo la macchina Diag:

Diag accetta (yes) se H dice no; ovvero se M(w) =no

Diag rigetta (no) se H dice si; ovvero se M(w) =s

Diag:



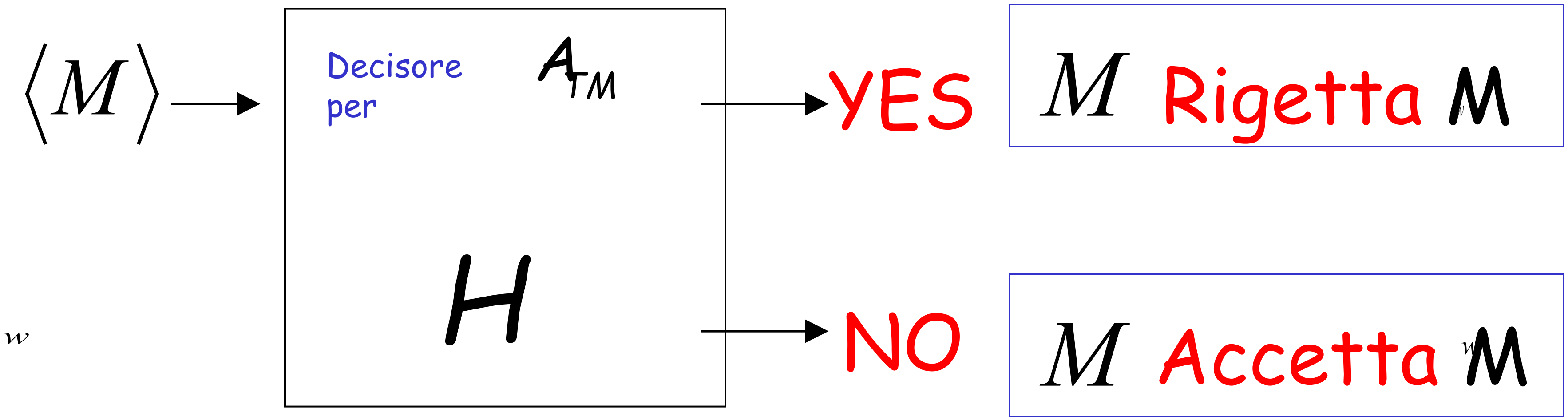
Semplifichiamo Diag .

Definiamo Diag

Descrizione di Diag:

Diag accetta M se M rigetta M

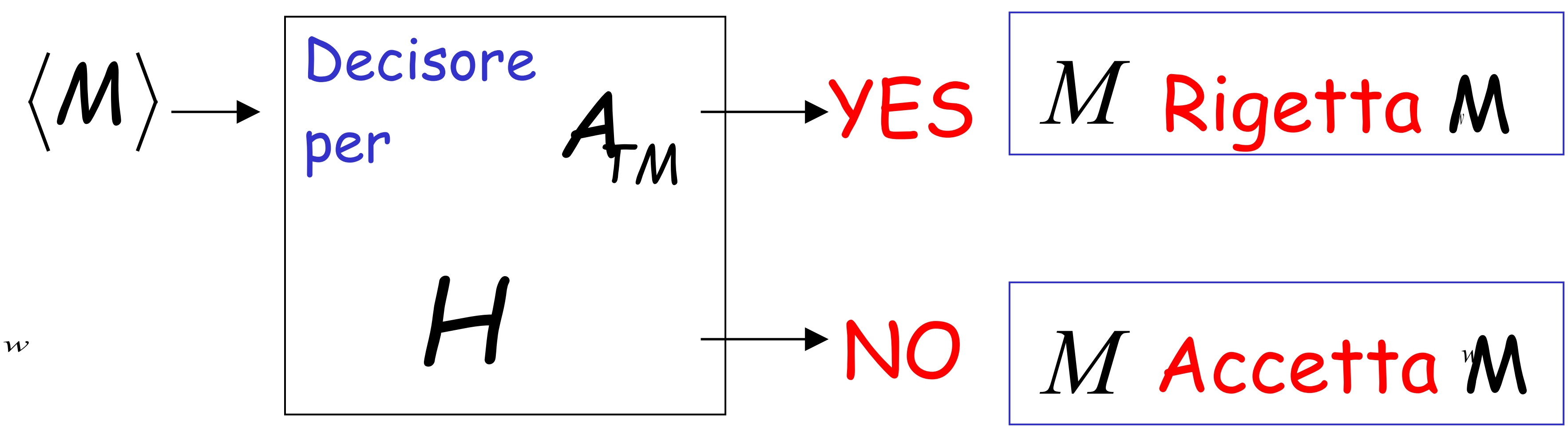
Diag rigetta M se M accetta M



Descrizione di Diag:

Diag accetta (yes) Diag se Diag rigetta Diag (no)

Diag rigetta (no) Diag se Diag accetta Diag (yes)



Descrizione di D:

Diag accetta M se M rigetta M

Diag rigetta M se M accetta M

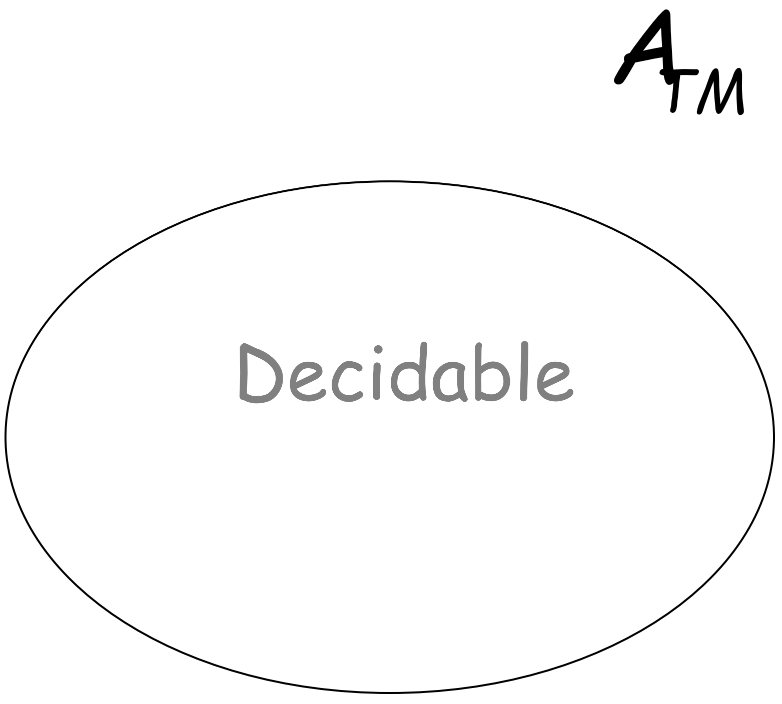
Al posto di M sostituiamo Diag

Cosa accade?:

Diag accetta Diag se Diag rigetta Diag (!!)

Diag rigetta Diag se Diag accetta Diag (!!)

Abbiamo mostrato:



Definizione di Turing accettabile

M accetta D se M(D) raggiunge uno stato finale.

Nulla è stato detto su cosa accade quando M rifiuta D

$$A_{TM} = \{\langle M, w\rangle : M \text{ è una Turing machine che accetta la stringa } w\}$$

A_{TM} è Turing-Acceptable (semidecidibile)

Turing machine che accetta A_{TM}:

$\langle M,w \rangle \longrightarrow$

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

domanda: M si ferma nel processo di calcolo con stringa di input w?

linguaggio corrispondente:

$$HALT_{TM} = \{\langle M, w\rangle : M \text{ è una Turing machine che si ferma sull'input w }\}$$

$$A_{TM} = \{\langle M, w\rangle : M \text{ è una Turing machine che accetta la stringa } w\}$$

$$HALT_{TM} = \{\langle M, w\rangle : M \text{ è una Turing machine che si ferma sull'input w }\}$$

Teorema: $HALT_{TM}$ è indecidibile

(The halting problem non è risolvibile)

dim:

idea di base:

Supponiamo che $HALT_{TM}$ è decidibile;

Proveremo che:

ogni linguaggio Turing-Acceptable

è decidibile

contradizione!

Supponiamo che $HALT_{TM}$ è decidibile

Input string $\langle M, w \rangle$

$\langle M \rangle \rightarrow$ Decisore per $HALT_{TM}$ $\rightarrow$ YES: $M\ w \downarrow$ alt

$w \rightarrow$ Decisore per $HALT_{TM}$ $\rightarrow$ NO: $M\ w \uparrow$ No alt sia L un linguaggio Turing-Accettabile

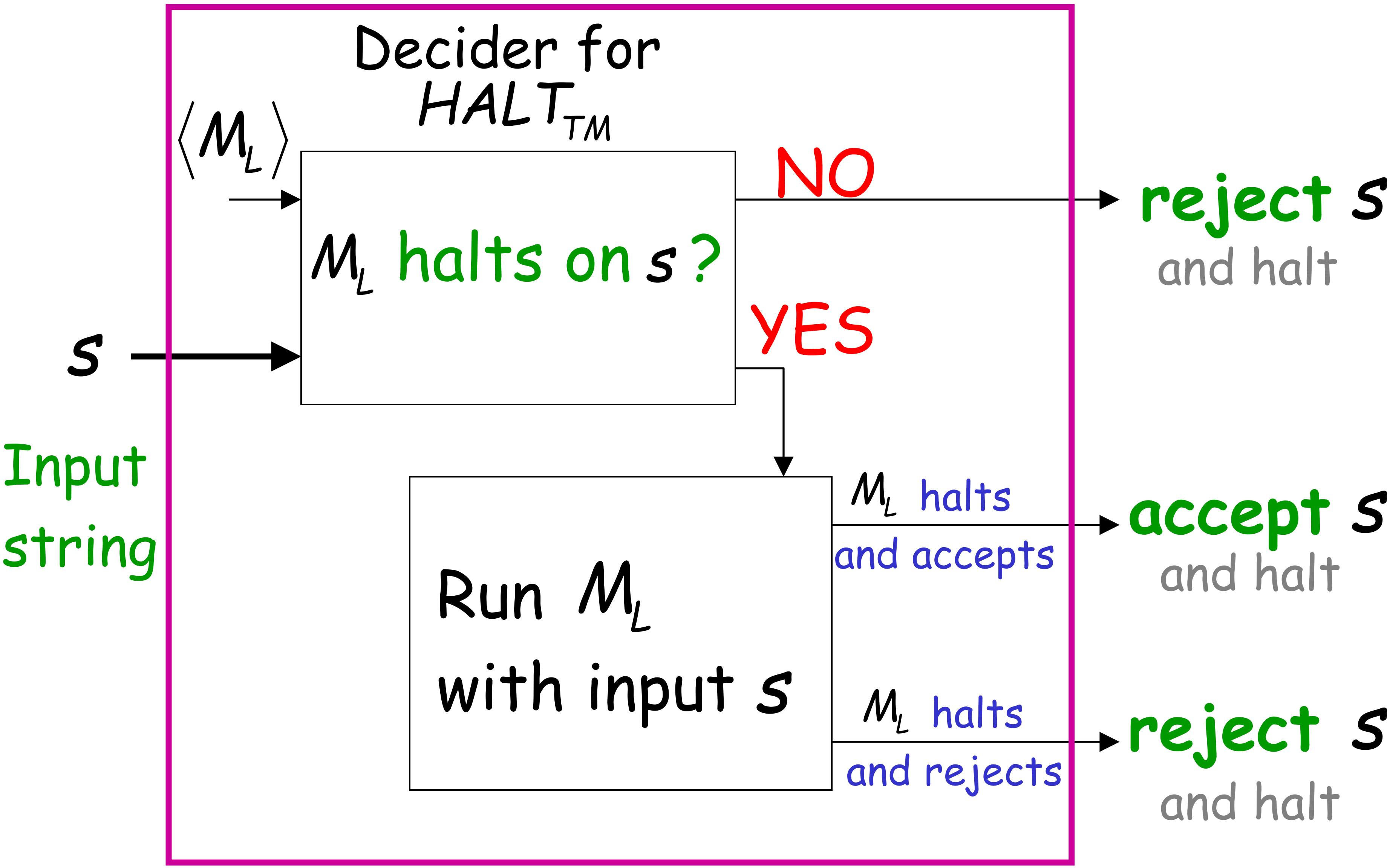
Turing-Semidecidibile

sia M_L la Turing Machine che accetta L

Proviamo che L è decidibile:

Costruiamo un decisore per L

Decider per L



Quindi L è decidibile

poichè L è stato scelto arbitrariamente, ogni linguaggio (Turing-) semidecidibile è decidibile

Ma vi è un linguaggio Turing-Accettabile (semidecidibile) che è indecidibile (Teor, visto in precedenza)

Contradizione!!!!

END OF PROOF

Uno sguardo sulla diagonalizzazione

Un altra dimostrazione

Teorema: $HALT_{TM}$ è indecidibile

(The halting problem non è decidibile)

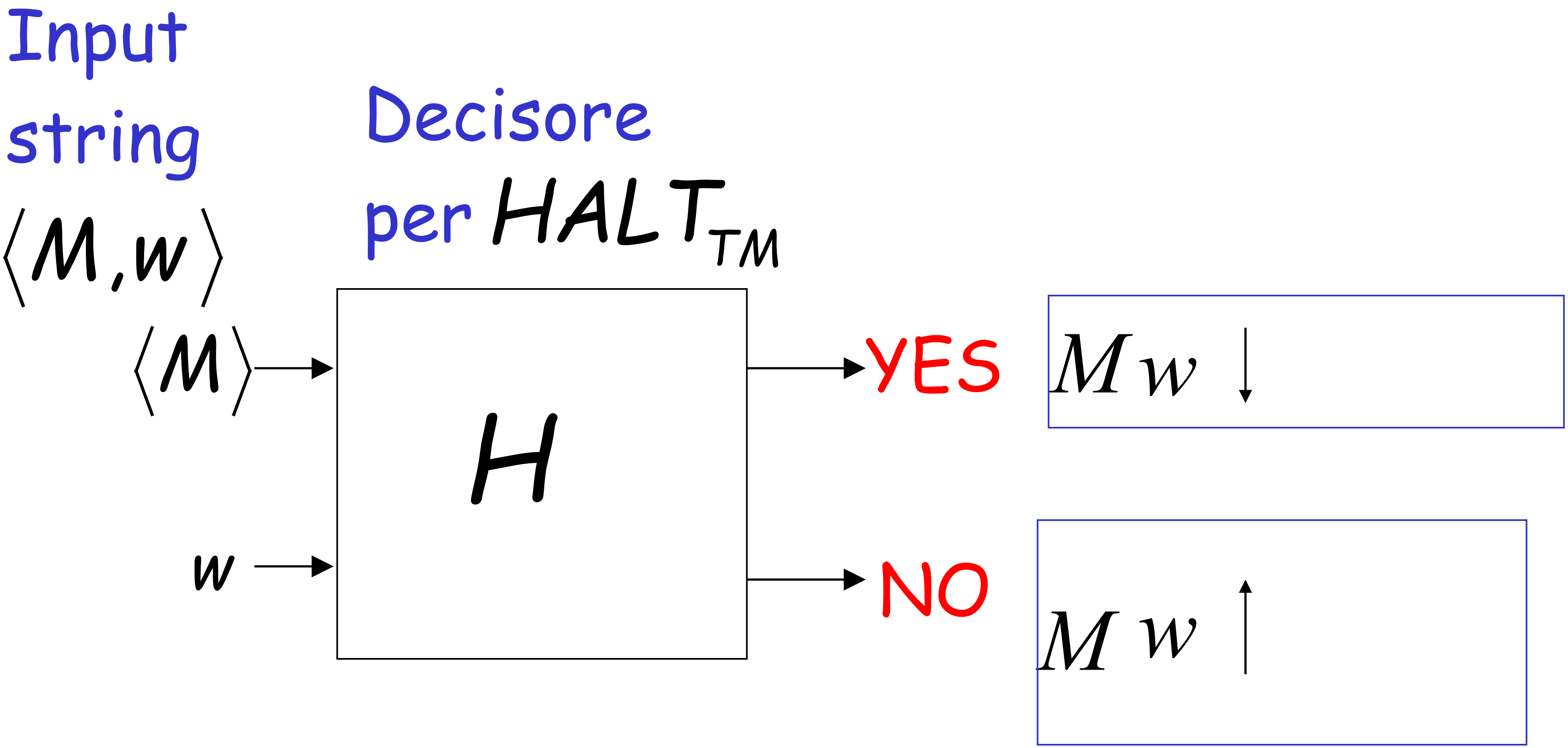
dim:

Idea di base:

Per assurdo: assumiamo che l' halting problem decidibile;

Cercheremo di ottenere una contradizione via diagonalizzazione

Supponi che $HALT_{TM}$ è decidibile

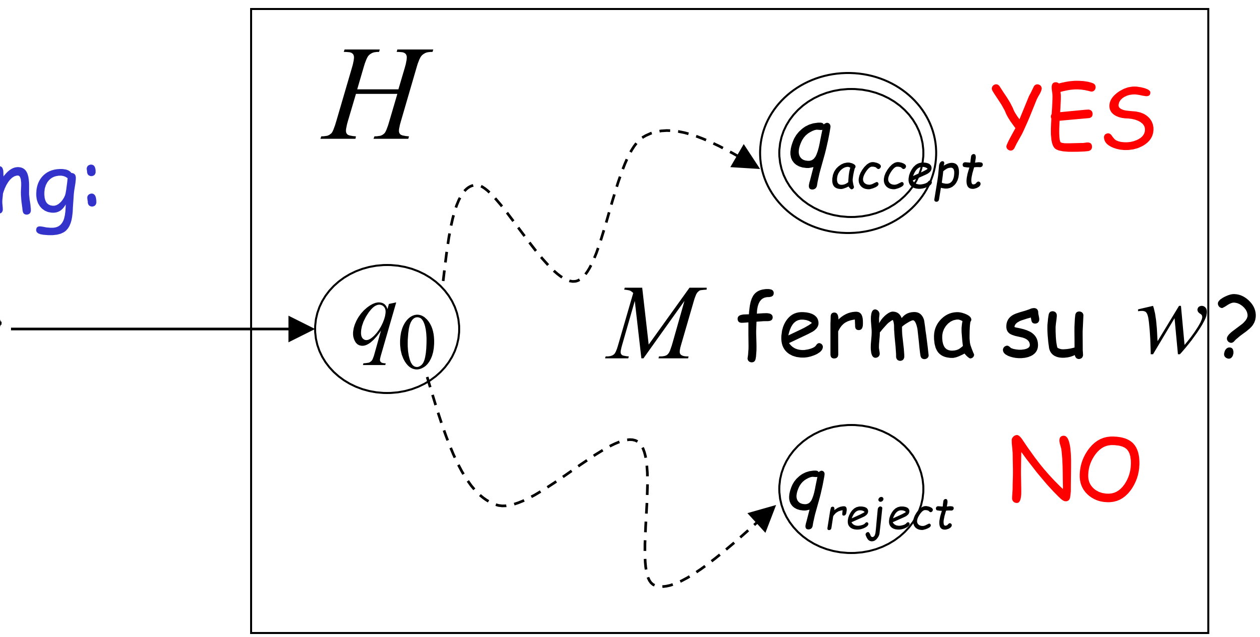


Guardiamo dentro H

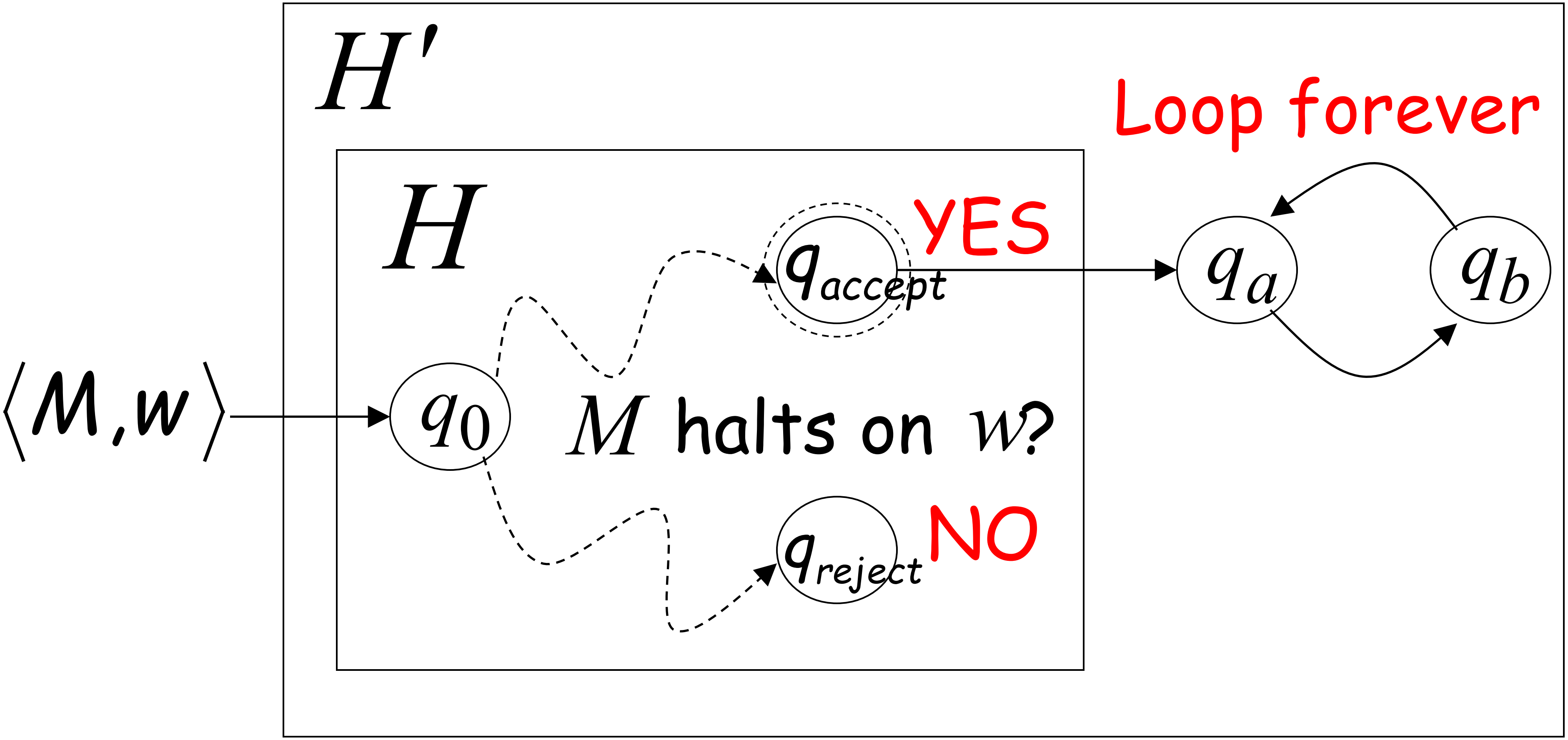
Decider per $HALT_{TM}$

Input string:

$\langle M, w \rangle$



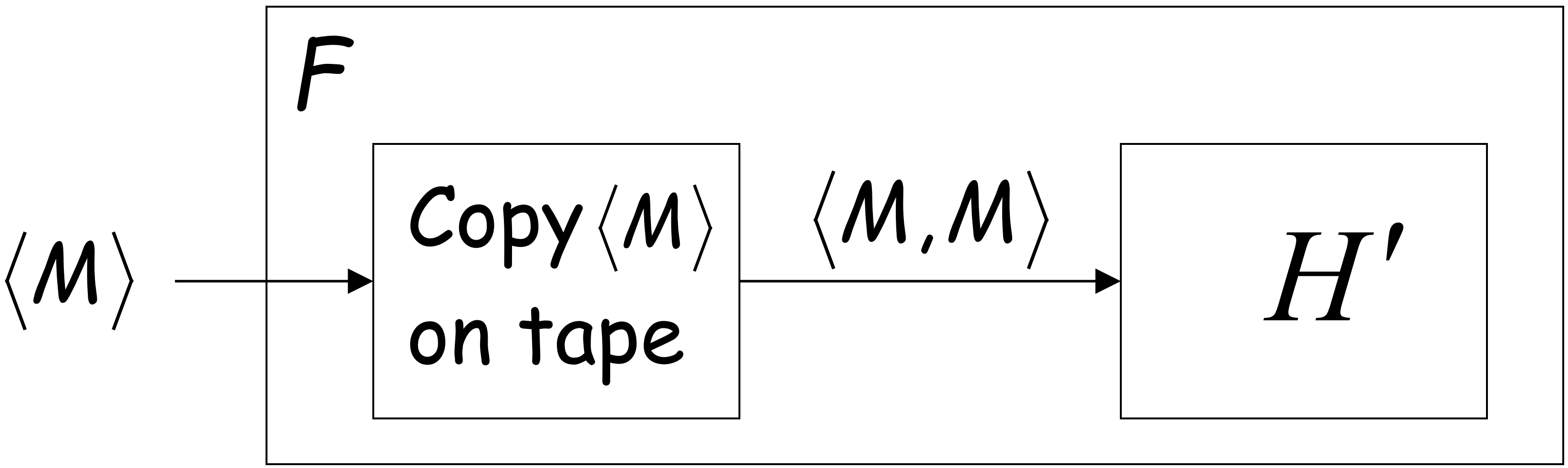
Costruiamo la macchina H' :



If M halts on input w **Then** Loop Forever

Else Halt

Costruiamo la macchina F:

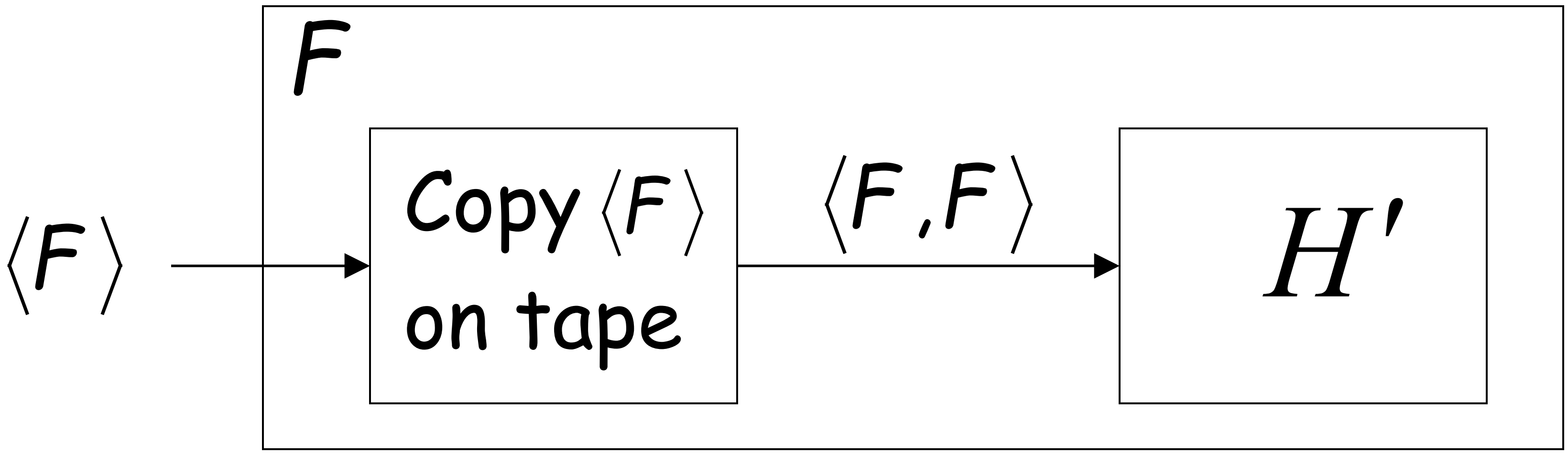


If M halts on input $\langle M \rangle$

Then loop forever

Else halt

calcola F con input se stesso



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

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

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

contradizione!!!

END OF PROOF

Idem precedente

Teorema 5.8 *[Indecidibilità del problema della terminazione[8]] Siano dati un alfabeto Γ ed una codificazione che associa ad ogni macchina di Turing $\mathcal{M} = \langle \Gamma, \not{b}, Q, \delta, q_0, F \rangle$ una sua codifica $c_{\mathcal{M}} \in \Gamma^*$. La funzione*

$$h(c_{\mathcal{M}}, x) = \begin{cases} 1 & \text{se } \mathcal{M} \text{ termina su input } x \\ 0 & \text{se } \mathcal{M} \text{ non termina su input } x \end{cases}$$

non è T-calcolabile.

$h\,(\text{codice}_M, x)$ =1 se M con input x termina
=0 se M con input x non termina

Supponiamo che il predicato sia calcolabile, esista cioè una macchina di Turing h che calcola la funzione h. Costruiamo la macchina h' che calcola il predicato

$h'(\text{codice}_M)$ =1 se M con input Codice_M termina
=0 se M con input Codice_M non termina

h' è la composizione di due macchine:

la prima con input $codice_M$ fornisce $codice_M$ b $codice_M$,

la seconda è la macchina h che calcola il predicato della terminazione.

In altre parole h' è la macchina che verifica se una MT termina quando le viene fornito in input il proprio codice.

Possiamo ora costruire una nuova macchina h'' che prende in input $codice_M$ e calcola la funzione:

$$h''(codice_M) \quad = 0 \text{ se } h'(codice_M) = 0$$
$$= \text{indefinito altrimenti}$$

h'(codice$_M$) **= 1** **se M (codice$_M$) termina**
= 0 **se M(codice$_M$) non termina**

h''(codice$_M$) = 0 se h'(codice$_M$) = 0 **(se M(codice$_M$) non termina)**
= **indefinito altrimenti (se M (codice$_M$) termina)**
)

termina con 0 se h' si è fermata con 0 e
si mette a ciclare, se h' si è fermata con 1

calcoliamo $h''(codice_{h''})$:

$h''(codice_{h''}) =$ indefinito se $h''(codice_{h''})$ è definita
$= 0$ se $h''(codice_{h''})$ è indefinita

In ogni caso abbiamo una contraddizione. Quindi non può esistere la macchina H.

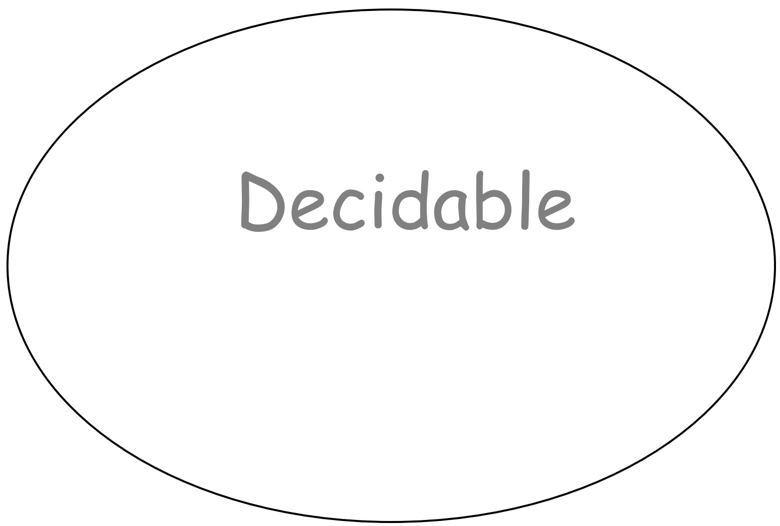
$h''(D_M) = 0$ se $h'(D_M) = 0$
$=$ indefinito altrimenti

Fine
idem

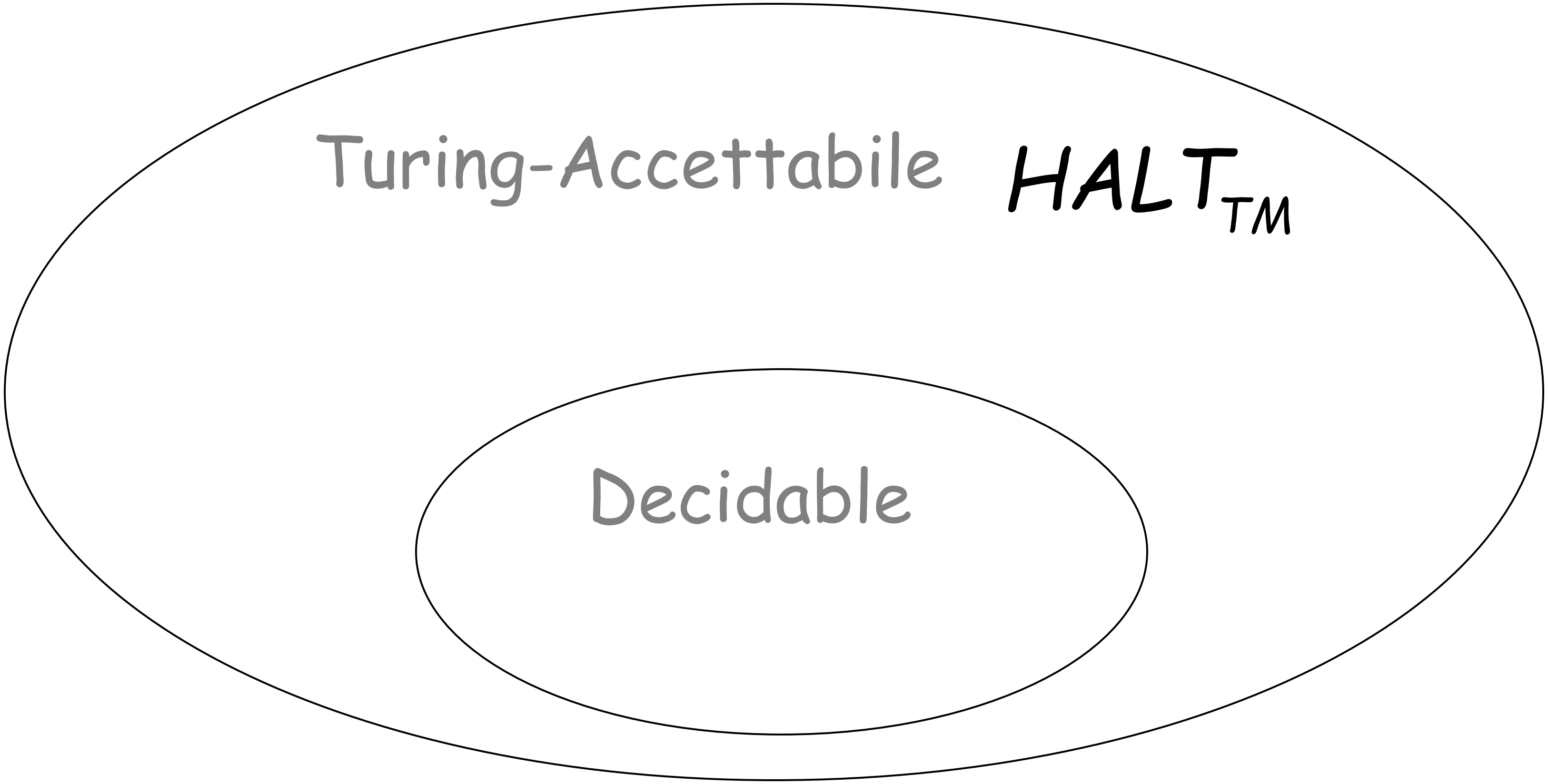
$h'(D_M) = 1$ se $M(D_M)$ termina
$= 0$ se $M(D_M)$ non termina

Abbiamo mostrato

indecidibile $HALT_{TM}$



Adesso proviamo che:



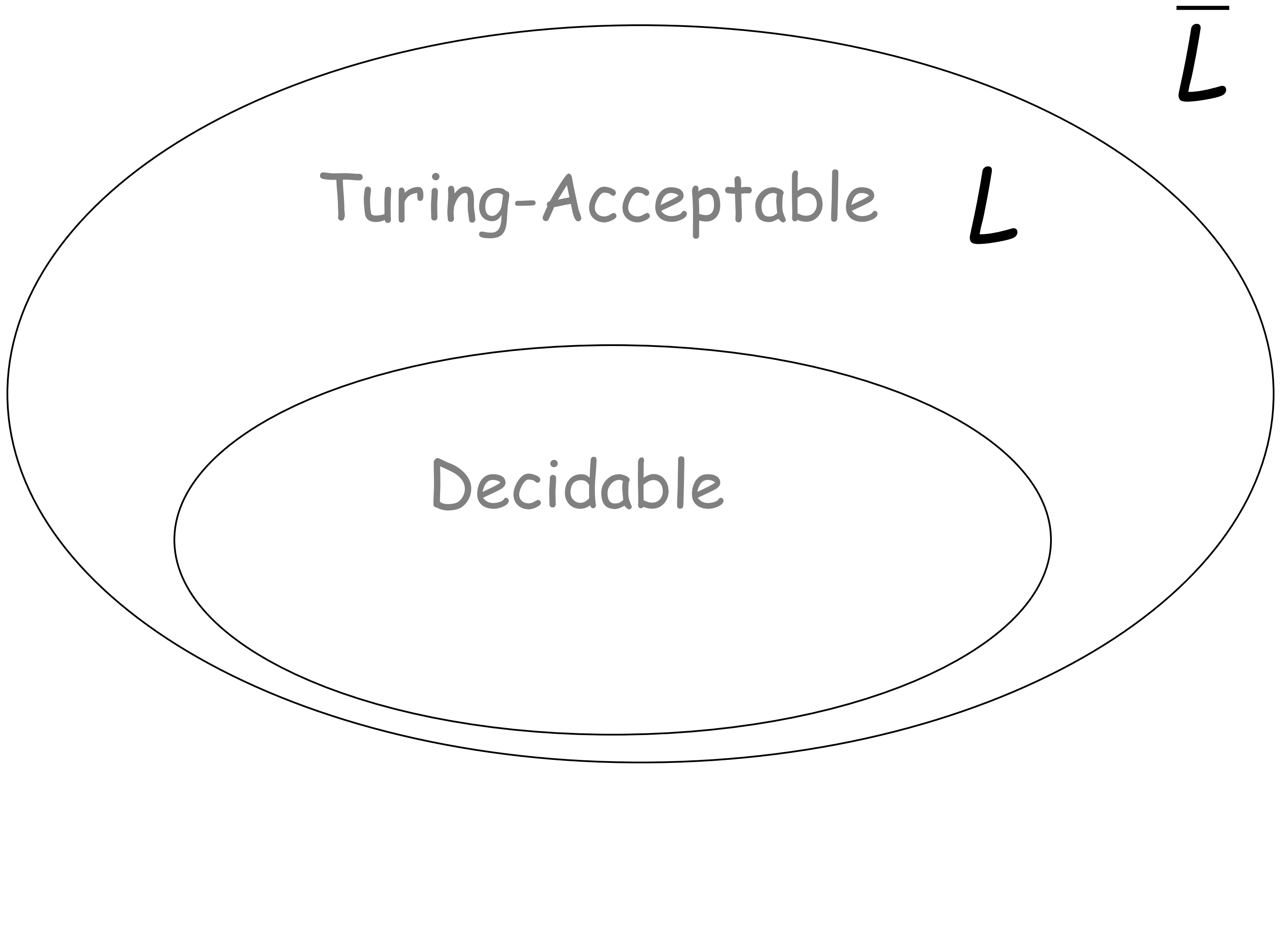
$HALT_{TM}$ è Turing-Acceptable (semidecidibile)

Turing machine che accetta $HALT_{TM}$:

$\langle M,w \rangle \longrightarrow$

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

Abbiamo gia mostrato che esistono linguaggi indecidibili:

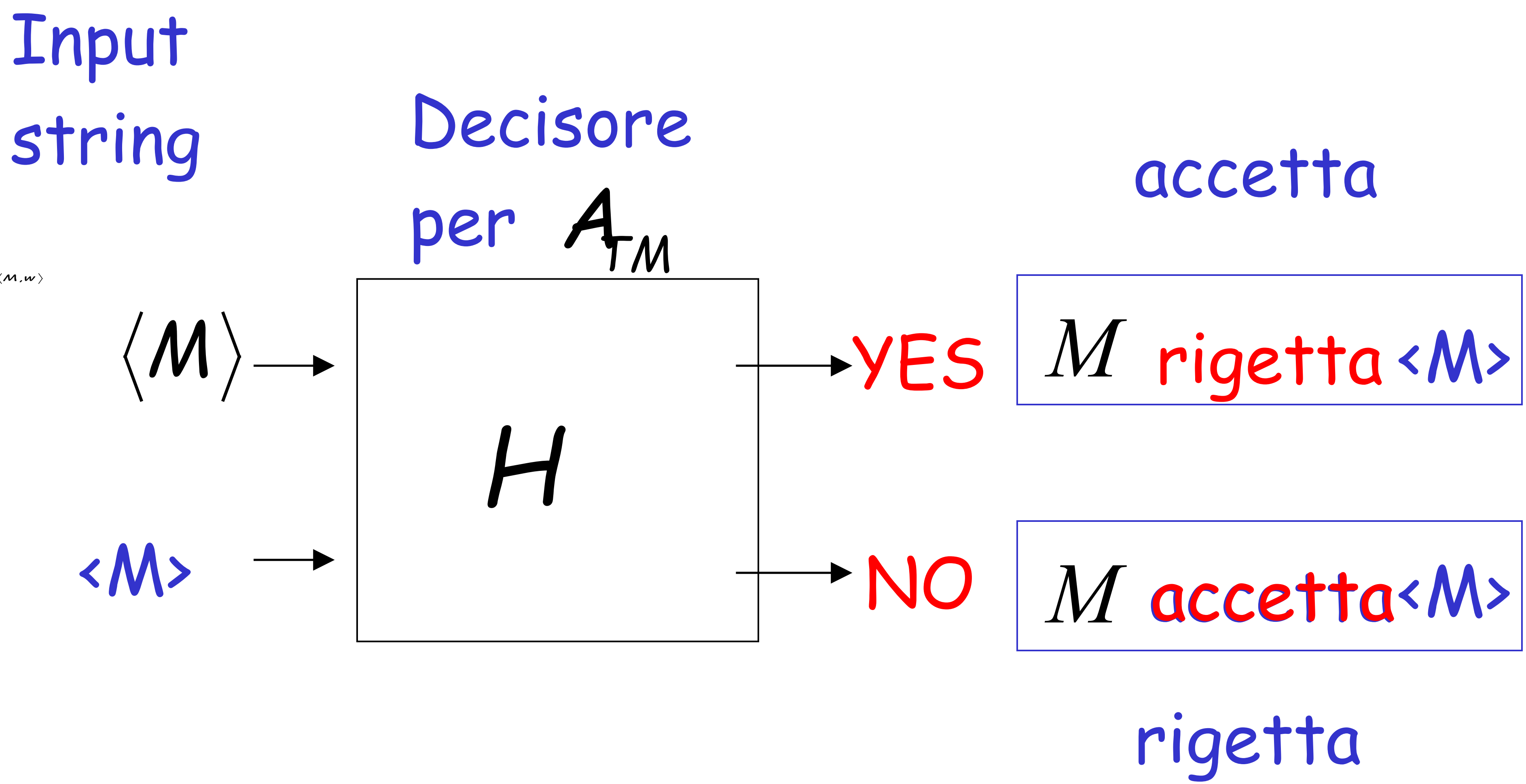


Tesi di Church Turing

Fine indecidibilità.

A_{TM}

Supponiamo w=<M> e
calcoliamo D(<M> , <M>)



Teorema: A_{TM} è indecidibile

(The membership problem non è risolvibile)

Proof:

Idea di base:

Assumiamo che A_{TM} è decidibile;

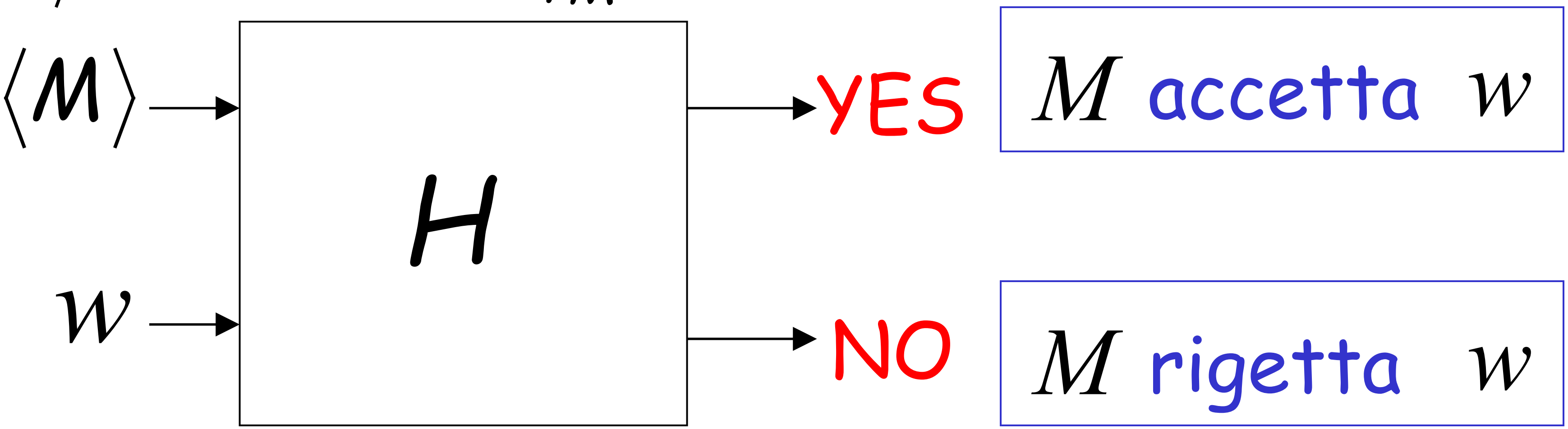
Proveremo che ogni linguaggio è Turing-Acceptable

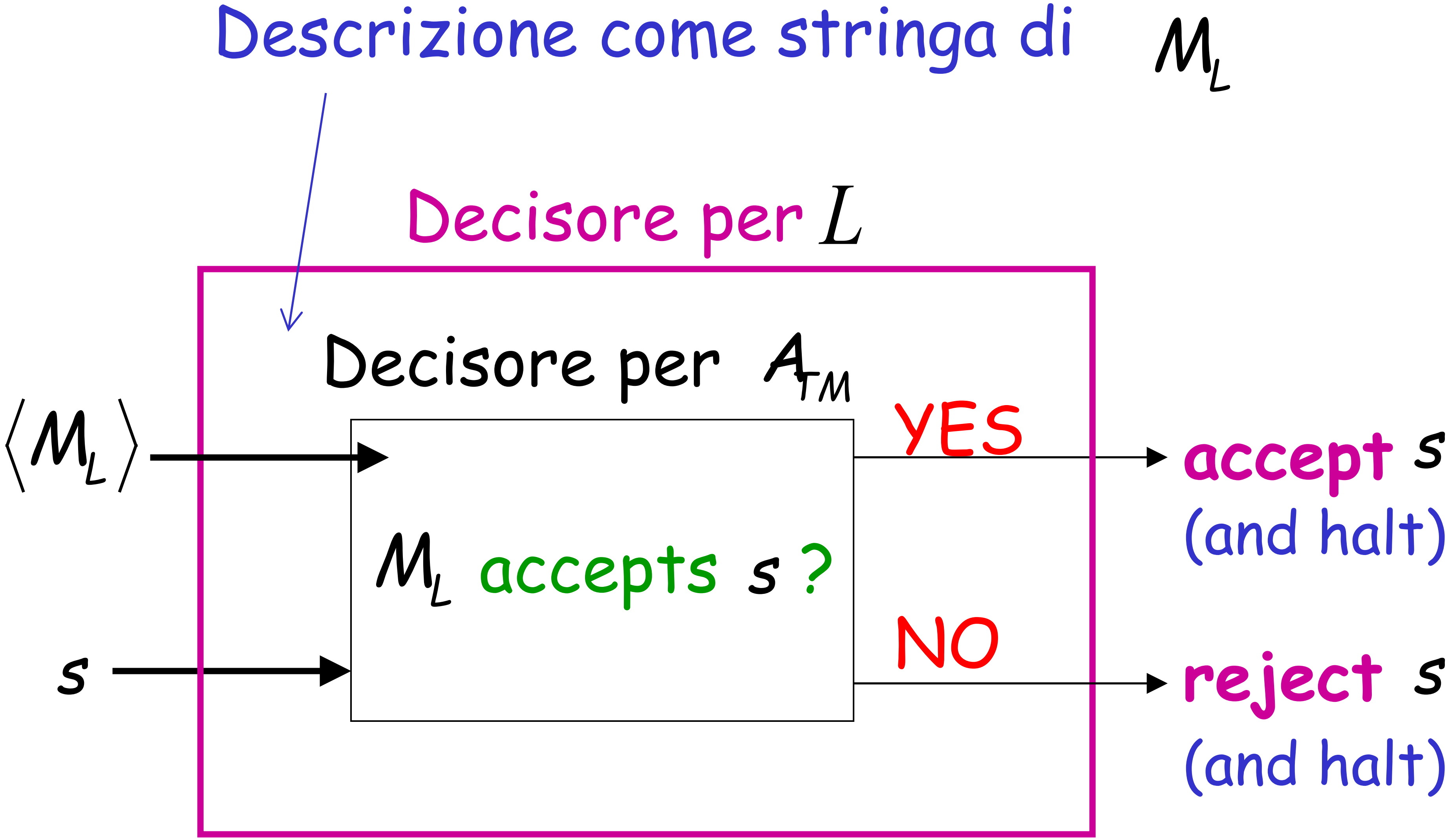
assurdo!

Allora avremo un decisore cosi definito:

Input string $\langle M,w \rangle$

Decisore per A_{TM}





Sia M_L il decisore per il linguaggio L

Cambiamo, diagonalizziamo, chiamiamo la macchina Diag.

Diag accetta se A_{TM} dice yes ovvero se M(w) =no

Diag rigetta se A_{TM} dice no ovvero se M(w) =s

accetta

Diag:

$\langle M,w \rangle$

$\langle M \rangle \rightarrow$ Decisore per A_{TM} H $\rightarrow$ YES M rigetta w

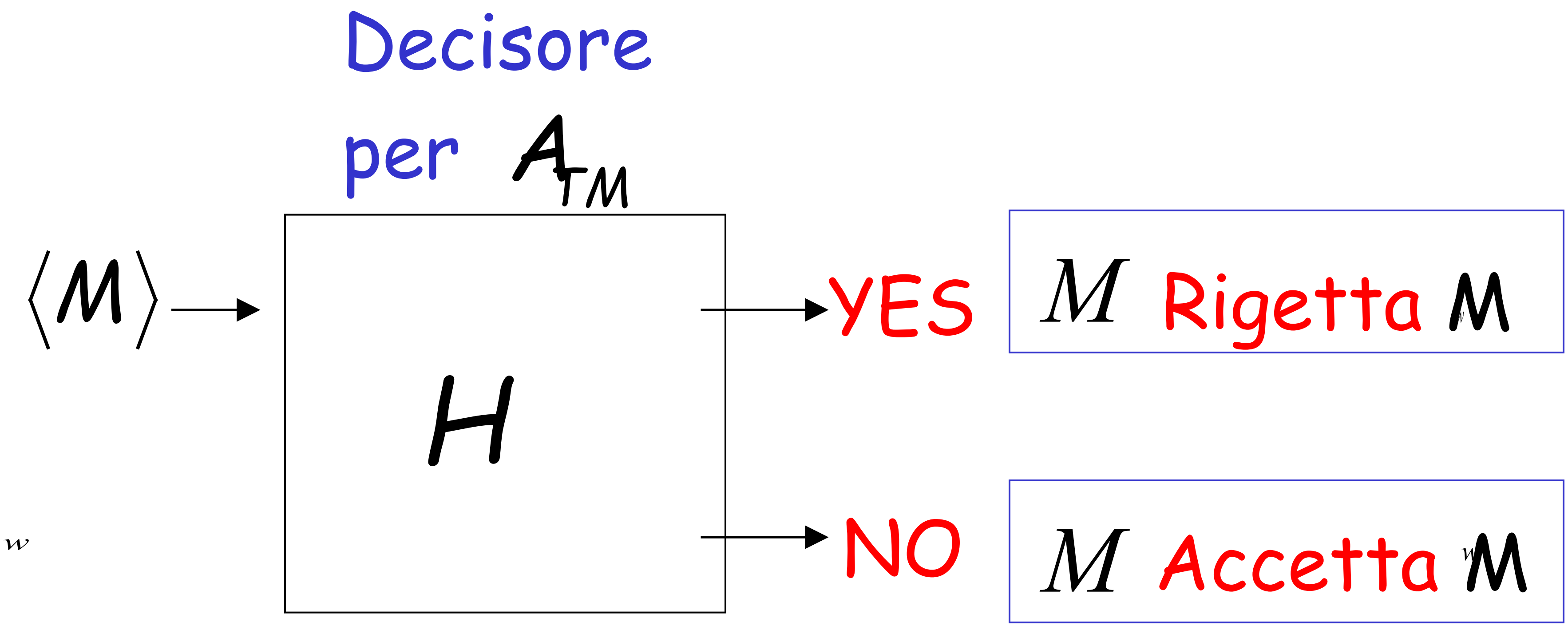
$w \rightarrow$ $\rightarrow$ NO M accetta w

rigetta

Descrizione di Diag:

Diag accetta M se M rigetta M

Diag rigetta M se M accetta M



Mostriamo che: