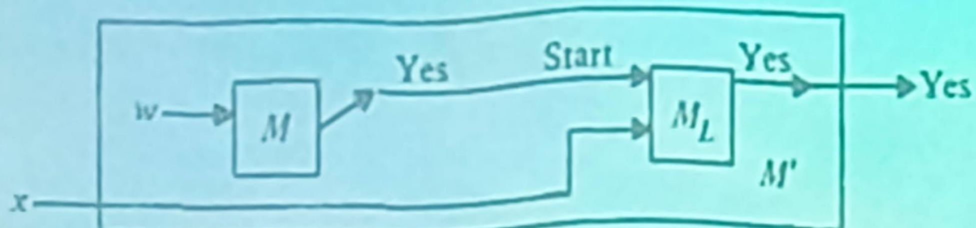
Theorem 8.6 (*Rice's Theorem*) **Any nontrivial property $\mathcal{S}$ of the r.e. languages is undecidable.**

$\boxed{L_u \leq L_{\mathcal{S}}}$ L_u is not recursive $\Rightarrow$ $L_{\mathcal{S}}$ is not recursive

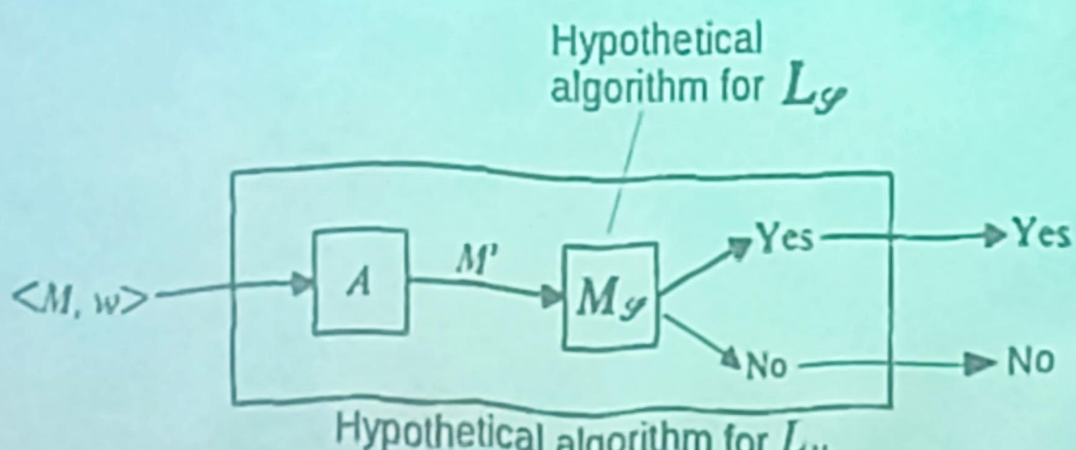
We need an algo A s.t. $A(M, w) = \langle M' \rangle$ s.t. $\begin{cases} L(M') \in \mathcal{S} & \text{if } w \in L(M) \\ L(M') \notin \mathcal{S} & \text{if } w \notin L(M) \end{cases}$

Wlog assume $\phi \notin \mathcal{S}$. Take any r.e. language $L \in \mathcal{S}$. $\exists$ TM M_L s.t. $L = L(M_L)$.



$L(M') = L \in \mathcal{S}$ if $w \in L(M)$
$L(M') = \phi \notin \mathcal{S}$ if $w \notin L(M)$

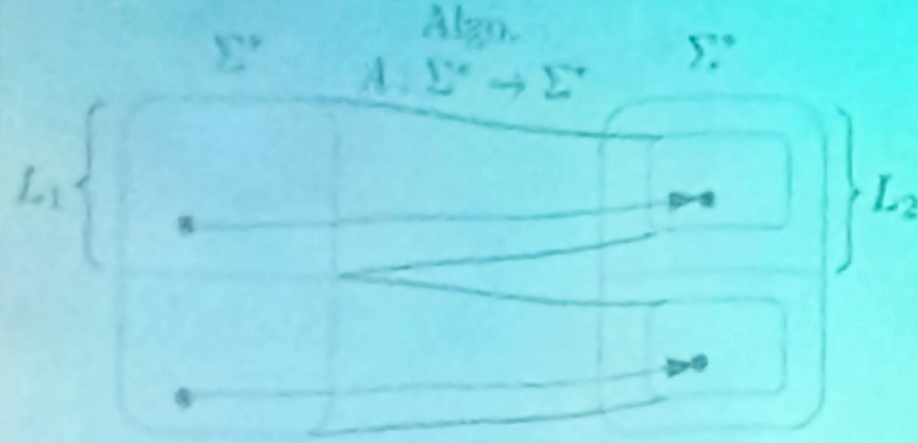
Fig. 8.11 M' **used in Rice's theorem.**



For $L_1, L_2 \subset \Sigma^*$

A (many-one turing) reduction of L_1 to L_2 is an algothim $A : \Sigma^* \to \Sigma^*$

s.t. $x \in L_1$ if and only if $A(x) \in L_2$



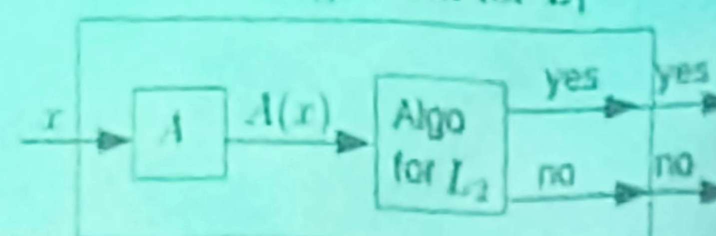
L_1 redudes to L_2 if there exists a reduction

$L_1 \leq L_2$

If $L_1 \leq L_2$ then,

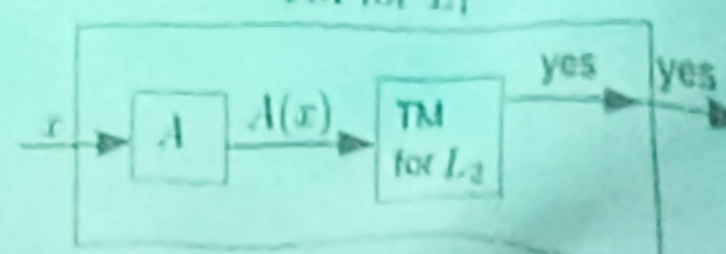
L_2 is recursive implies L_1 is recursive

Algorithm for L_1



If $L_1 \leq L_2$ then, L_2 is r.e. implies L_1 is r.e.

TM for L_1



M is a TM $\qquad$ L(M) is r.e.

Construct a TM M' with output

$x_1 \# x_2 \# x_3 \#$ ---------- may not stop

x_i's may not be distinct.

x_i appears in the list (eventually) iff $x_i \in L(M)$