More Reductions

Recall :

For any two languages $\mathcal{L}$ and R over alphabets Σ_1 and Σ_2,
if there is a total and computable function $\sigma : \Sigma_1^* \to \Sigma_2^*$ s.t.
for any $\omega \in \Sigma_1^*$, $\omega \in \mathcal{L}$ iff $\sigma(\omega) \in R$, then,
we say that $\mathcal{L}$ reduces to R (denoted $\mathcal{L} \leq R$), and
if $\mathcal{L}$ is (independently shown to be) undecidable, so is R

$\mathcal{L} \leq R$: R is at least as difficult (to decide) as $\mathcal{L}$

If $\mathcal{L}$ is undecidable, so is R

Claim: $\mathcal{L}_\varepsilon = \{\langle M \rangle \mid M$ is a TM and M accepts $\varepsilon\}$ is undecidable.

① Suppose $\mathcal{L}_\varepsilon$ is decidable. There is a machine E deciding $\mathcal{L}_\varepsilon$.

② Want to show $\mathcal{L}_{TM} \leq \mathcal{L}_\varepsilon$.

$\sigma(s) = \langle P_s \rangle$, where P_s does the following on input x:

(a) If s is of the form $\langle M \rangle \# \omega$ where M is a TM, proceed to (b).

(b) Run M on ω, accept x if M accepts ω.

x is irrelevant!

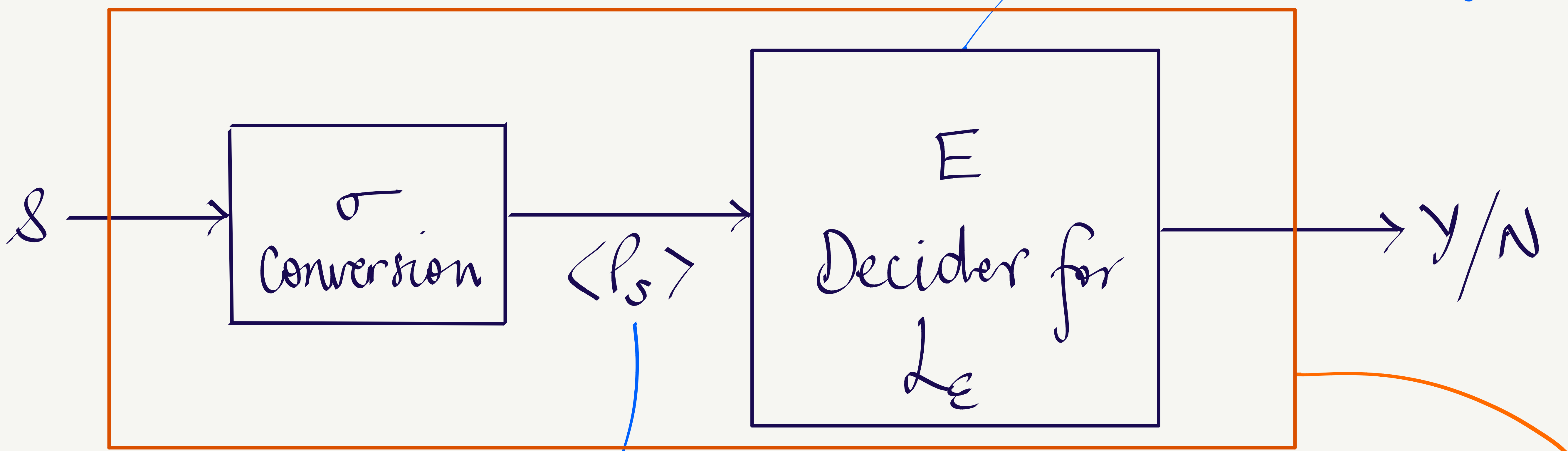
$$s \in \mathcal{L}_{TM} \Rightarrow \mathcal{L}(P_s) = \Sigma^* \Rightarrow \varepsilon \in \mathcal{L}(P_s) \Rightarrow \langle P_s \rangle \in \mathcal{L}_\varepsilon$$

$$s \notin \mathcal{L}_{TM} \Rightarrow \mathcal{L}(P_s) = \phi \Rightarrow \varepsilon \notin \mathcal{L}(P_s) \Rightarrow \langle P_s \rangle \notin \mathcal{L}_\varepsilon$$

So, $s \in \mathcal{L}_{TM} \Leftrightarrow \sigma(s) \in \mathcal{L}_\varepsilon$.

Creating $\langle P_s \rangle$ is easy; it does not involve actually running P_s (or M)!

E accepts if P_s accepts ε,
rejects if P_s does not accept ε



This machine decides L_{TM}

P_s accepts any string if s is of the form $\langle M \rangle \# w$,
where M is a TM and M accepts w;
otherwise, P_s does not accept <u>any</u> string

But L_{TM} is undecidable, so this yields a contradiction!
So our assumption about the decidability of L_ε was wrong.
$\therefore$ L_ε is undecidable.

What about the following language?

$$\mathcal{L}_S = \{\langle M\rangle \mid M \text{ accepts } \langle M\rangle\}$$

Claim: $\mathcal{L}_S = \{\langle M\rangle \mid M \text{ accepts } \langle M\rangle\}$ is undecidable.

① Suppose $\mathcal{L}_S$ is decidable. There is a machine S deciding $\mathcal{L}_S$.

② Want to show $\mathcal{L}_{TM} \le \mathcal{L}_S$.

$\sigma(s) = \langle P_s\rangle$, where P_s does the following on input x:

(a) If s is of the form $\langle M\rangle\#\omega$ where M is a TM, proceed to (b).

(b) Run M on ω, accept x if M accepts ω.

$$s \in \mathcal{L}_{TM} \Rightarrow \mathcal{L}(P_s) = \Sigma^* \Rightarrow \langle P_s\rangle \in \mathcal{L}(P_s) \Rightarrow \langle P_s\rangle \in \mathcal{L}_S$$

$$s \notin \mathcal{L}_{TM} \Rightarrow \mathcal{L}(P_s) = \emptyset \Rightarrow \langle P_s\rangle \notin \mathcal{L}(P_s) \Rightarrow \langle P_s\rangle \notin \mathcal{L}_S$$

So, $s \in \mathcal{L}_{TM} \Leftrightarrow \sigma(s) \in \mathcal{L}_S$.

But $\mathcal{L}_{TM}$ is undecidable. So contradiction! $\mathcal{L}_S$ is undecidable.