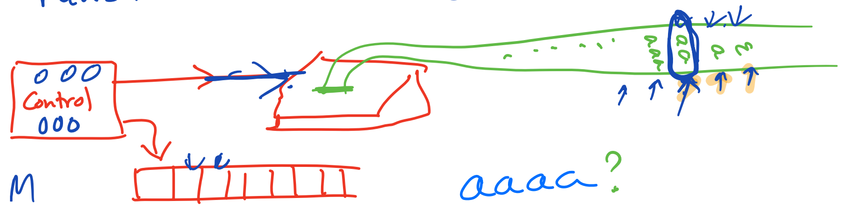
Enumerator TM

An **enumerator TM** is a variant of TM that is attached to a printer.

- Starts with a blank tape
- runs ... will occasionally print a string (on printer)



aaaa?

The **language generated** by an enumerator TM is the set of strings it prints out.

- each string is printed out after a finite number of other strings are printed.

"recursively enumerable"

Theorem 3.21 A language is $\underbrace{\text{Turing recognizable}}$ iff some enumerator-TM enumerates it.

Proof:

($\Leftarrow$) Suppose we have an enumerator TM E.

We construct a TM M that recognizes $L(E)$

if $\notin L(E)$, then M will loop forever.

M = "on input w :

1. Run E.
2. $\forall$ w' that E prints out, compare w' to w.

   2.1 if $w = w'$, ACCEPT."

ε
aaa
aa
aaaaa
ab
ba
aaaaa
⋮

($\Rightarrow$) Suppose we have a recognizer M for some language L(M).
We construct an enumerator TM for L(M).

Let $s_1, s_2, s_3, \ldots$ be the strings of Σ^* in shortlex order.

$\{a, b\}$.

E = " ignore the input.

1. Let $i = 1$.

1.1 Run M on input s_i

1.2 if M accepts s_i, PRINT s_i

if M rejects s_i don't print

1.3 $i = i+1$; go to 1.1

What is wrong with E?

on a M accepts

on b M loops

on aa ~~M~~ accept

Another possibility...

E = " ignore the input.

1. for $i = 1, 2, 3, \ldots$ Generate strings $s_1, \ldots, s_i$

1.1 Run M for i steps on each of

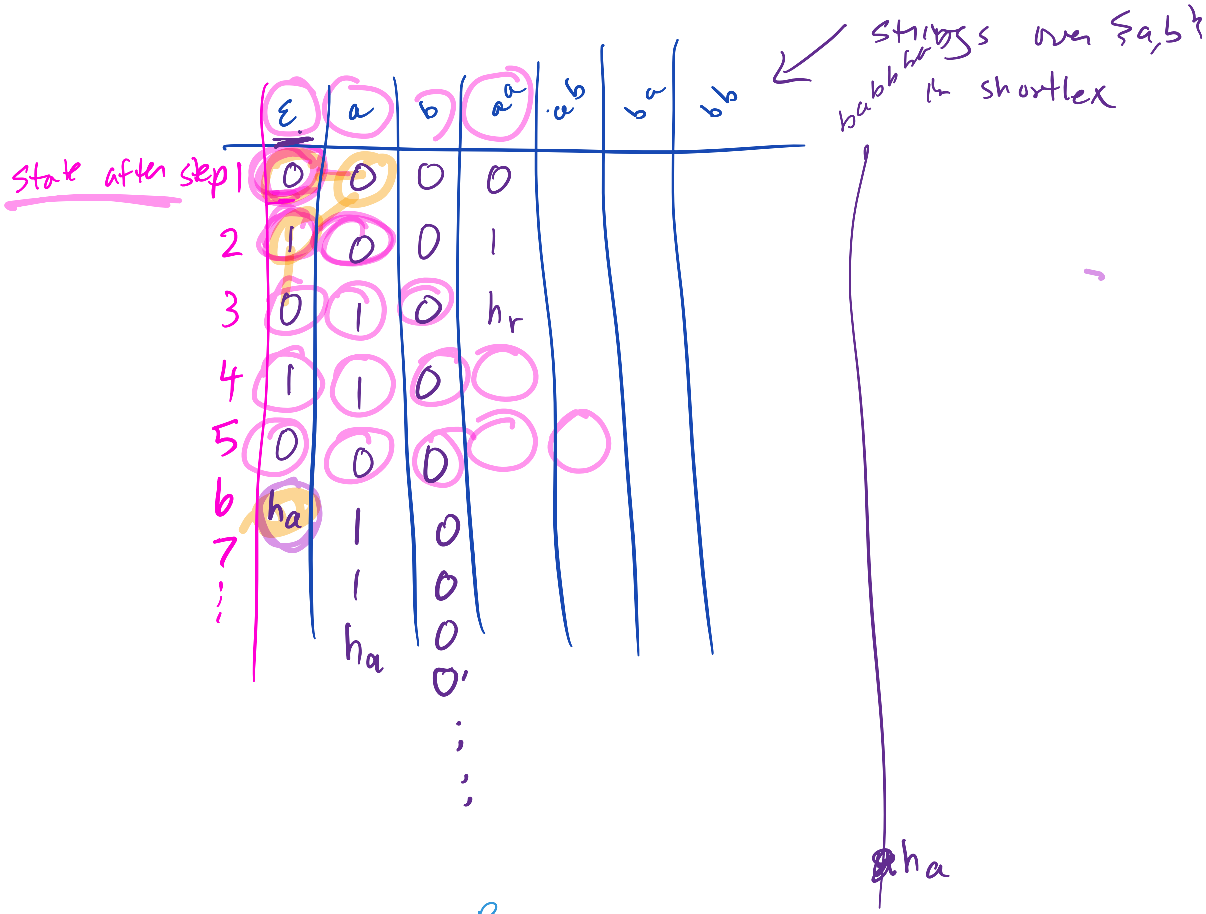
$s_1, s_2, \ldots, s_i$ ← strings over Σ in shortlex order.

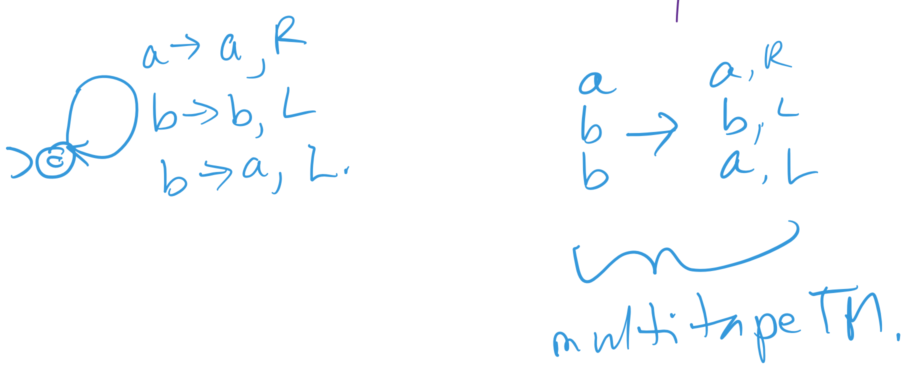
1.2 if any of the computations accept, print the corresponding string.

1.3 $i = i+1$; Go to 1.1 "

If S_{1000} is accepted by M after 2000 steps, then it will be printed by E when $i = 2000$. $(\max(1000, 2000))$.

Each string that M accepts will eventually be printed by E. $\blacksquare$





Claim: If $\exists$ a decider M for language L, then $\exists$ an enumerator for L.

Proof: Let M be a decider for L.

$E =$ " o: ignore the input.

1. - generate strings in shortlex order over Σ.

2. - for each such string w, run M on w.

   - if M accepts w, print w.
   - if M rejects, go to 1. "

Question: If $\exists$ an enumerator E for L. can we construct a <u>decider</u> for L?

Seems hard.

$\Uparrow$

Can we detect TM <u>loops</u>?