Membership problem: (MP): Give $\langle M \rangle$ and x,
check if M accepts x

$$MP = \{(\langle M \rangle, x) \mid x \in L(M)\}$$

Fact: MP is r.e

Use $\mathcal{U}$ to simulate M on x
and accept if M accepts x

Halting problem (HP): Given $\langle M \rangle$ and x
check if M halts on x
↳ enters t or r

Fact: HP is r.e

Again use $\mathcal{U}$ to simulate M on x
accept if M enters either t or r

Some properties of recursive & r.e languages

* A language L is co-r.e if $\overline{L}$ is r.e

Theorem: If L is both r.e and co-r.e, then L is recursive

Proof: Let M be the TM s.t $L(M) = L$
and M' be the TM s.t $L(M') = \overline{L}$

TM M^*: Simulate M and M' on x
one step at a time
if M accepts x, halt & accept
if M' accepts x, halt & reject

* M^* is total: $x \in L$ or $x \in \overline{L}$
so one of M or M' accepts

$\overline{HP} = \{(\langle M \rangle, x) \mid M \text{ does not halt on } x\}$

$\overline{HP}$ is co-r.e

Enumeration machines & r.e sets

Enumeration machines - Special types of TMs

* no input
* need not halt
* one i/p / work tape & one o/p tape
* Special print state, wherein it prints a string in o/p tape, erases it and then exits the print state

Theorem: A language L is r.e iff there is an enumeration machine for L

Time-sharing simulation of M.