Tutorial 7

1. Show that recursively enumerable languages are closed under union and Kleene star operations.

a) Union. Suppose L_1 and L_2 are r.e languages.

A_1 and A_2 are ~~algo~~ TM/algo for them.

Algo/TM for $L_1 \cup L_2$ be A.

$A(x)$:

for k from 0 to ∞...

Simulate A_1 on x for ~~k steps atmost~~ k steps

Simulate A_2 on x for k steps

If A_1 ~~and~~ or A_2 accepted x, then accept x

b) Kleene star. Suppose L is r.e language.

A is TM/algo for L.

Algo/TM for L^* be A_1.

$A_1(x)$:

for k from 0 to ∞:

for every partition of x: ($2^{|x|}$ partitions but no matter)

partition be $x_1, x_2 \dots x_i$.

Simulate A on x_1, x_2 ... x_i

each for ~~atmost~~ k steps.

If A accepted all $x_1, x_2 \dots x_i$, then accept x.

5. $L = \{(\langle M \rangle, q) \mid \exists w$ such that M on input w enters state $q\}$

Let A be algo/TM for L.

$M'(y)$: ~~Simulate M on x~~ M on x

If M halts, accepts

$M''(M, x)$:

If $A(M', \text{accept state})$ accepts, accept $(\langle M \rangle, x)$

rejects, reject $(\langle M \rangle, x)$

→ Note that if L is assumed to be recursive, A always reaches accept/reject.

And then M'' is TM for halting problem... ⇒ Hence L is not recursive

We've shown $HP \leq_m L(M', \text{accept})$

If $(\langle M \rangle, x) \in HP$ then ~~$L(M') = \Sigma^*$~~ $(\langle M' \rangle, \text{accept}) \in L$

If $(\langle M \rangle, x) \notin HP$ then ~~$L(M') = \emptyset$~~ $(\langle M' \rangle, \text{accept}) \notin L$

Is L recursively enumerable?

Yes I can create Algo/TM for it.

$L \le_m HP \Rightarrow L$ is r.e.
$HP \le_m L \Rightarrow L$ is not r.e

$A(\langle M \rangle, q)$:
  for k from 0 to ∞
    for each w of first k string (lexicographical order)
      Simulate M for k steps.
      If M reached q anywhere, then accept.

6. $L = \{\langle M \rangle \mid M \text{ is total}\}$. Show L is not recursively enumerable.
Is L co-re?
Does undecidability of L follow from Rice Theorem.

Consider $\bar{L} = \{\langle M \rangle \mid M \text{ is not total}\}$.

Lemma: $\bar{L}$ is not r.e. (L is not co-re).

Suppose $\bar{L}$ is r.e. Then there is Algo/TM A accepting $\bar{L}$.
Create algo/TM A_1 for $\overline{HP}$ as follows.

$A_1(\langle M \rangle, x)$:
  Simulate A on M'
  If A accepts, then accept

$M'(y)$: Simulate M on x
  If M halts, accept.

→ If $(\langle M \rangle, x) \in \overline{HP}$, then M' loops for any input y, M' is not total, A accepts M'
A_1 accepts $(\langle M \rangle, x) \Rightarrow (\langle M \rangle, x) \in L(A_1)$

If $(\langle M \rangle, x) \notin \overline{HP}$, then M' halts for any input y, M' is total, A does not accept M'
$\Rightarrow (\langle M \rangle, x) \notin L(A_1)$

→ Hence $\overline{HP} \le_m \bar{L} \Rightarrow$ As $\overline{HP}$ is not r.e we can say $\bar{L}$ is not r.e
$\Rightarrow L$ is not co-re.

Lemma: L is r.e

Suppose L is r.e and there is Algo/TM A accepting L.
Create algo/TM for $\overline{HP}$ as follows:

$A_1(\langle M \rangle, x)$:
  Simulate A on M'
  If A accepts M', accept

$M'(k)$: Simulate M for k steps
  If M does not halt, accept
  If M halt loop here infinitely

→ $(\langle M \rangle, x) \in \overline{HP} \Rightarrow M'$ accepts any input k to it $\Rightarrow M'$ is total (always halt) $\Rightarrow A$ accepts M'
$\Rightarrow (\langle M \rangle, x) \in L(A_1)$

$(\langle M \rangle, x) \in HP \Rightarrow M'$ loops infinitely for some $k \Rightarrow M'$ not total $\Rightarrow A$ doesn't accept M'
$\Rightarrow (\langle M \rangle, x) \notin L(A_1)$

→ Hence $\overline{HP} \le_m L \Rightarrow L$ is not r.e

7. $L = \{\langle M \rangle \mid w \in L(M) \text{ iff } w^R \in L(M)\}$

Show that L is not recursive. Rice Theorum?

→ Clearly property is non-trivial here

→ Also $L(M_1) = L(M_2) \Rightarrow \langle M_1 \rangle \in L \Leftrightarrow \langle M_2 \rangle \in L$

So by Rice Thm, L is not recursive.

8. ~~Busy beaver function $BB : \mathbb{N} \to \mathbb{N}$~~

2. Consider a finite state automata similar to PDA but consists of a queue ~~stack~~ instead of a stack.

Show that if there is a TM accepting L, there is also a queue automata accepting L.

Will be informally stating why a queue automata accepting L exists.

→ Queue automata can simulate the turing machine tape

a b a a b # a ⊥ + states

queue

Can use # to denote head position. Place ~~#~~ after letter where head is.

To simulate Turing machine transition :-

→ Pop elements and push. ~~Remember~~ element last popped.

(Don't push element just popped).

→ When # encountered!

① ~~Place~~ The element the tape head is pointing to may need to be changed. Suppose to y. (y = unchanged possible)

② Tape head moving to LEFT: Push # first, Push y

③ Tape head moving to RIGHT: Push y, Push #.

→ When ⊥ encountered!

Push last element. Push ⊥.

Now the simulation of a turing machine transition is Done.

3. Let $L \subseteq \{0,1\}^*$ be any infinite r.e language. Show that L is recursive iff there exists an enumeration turing machine that lists the strings in L in lexicographical order.

($\Rightarrow$) If L is recursive there exists such an enumeration machine

→ Suppose algo/TM for L is A.

→ Build E as follows:

E: for w from 0 to ... lexicographically
Simulate A on w. (It will halt as L is recursive)
If A accepts w, print w

($\Leftarrow$) If exists E as mentioned above $\Rightarrow$ recursive L accepting
it. accepting
~~→ Suppose A is an algo/TM for L~~
→ Build A for L as follows:
$A(x)$:
Simulate E, (if E prints x then accepts
if E prints any string larger than x, reject)

4. Let $L_1 \subseteq \{0,1\}^*$ be any languages. Show that L_1 is r.e. iff
there is a recursive language L_2 such that
$L_1 = \{x \mid \exists y \text{ s.t } \langle x,y \rangle \in L_2\}$.