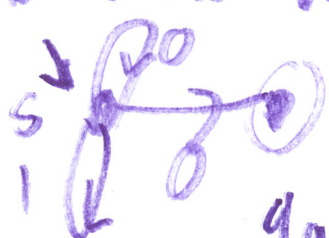
CSE 596 Lecture 10/3/18 Fall 2018

The trick of switching q_{acc} and q_{rej} does not affect running time or space, so it also shows that **EXP, PSPACE, P, L,** are closed under ~.

- Does not work for NTMs: Same as with an NFA:



q_{acc}

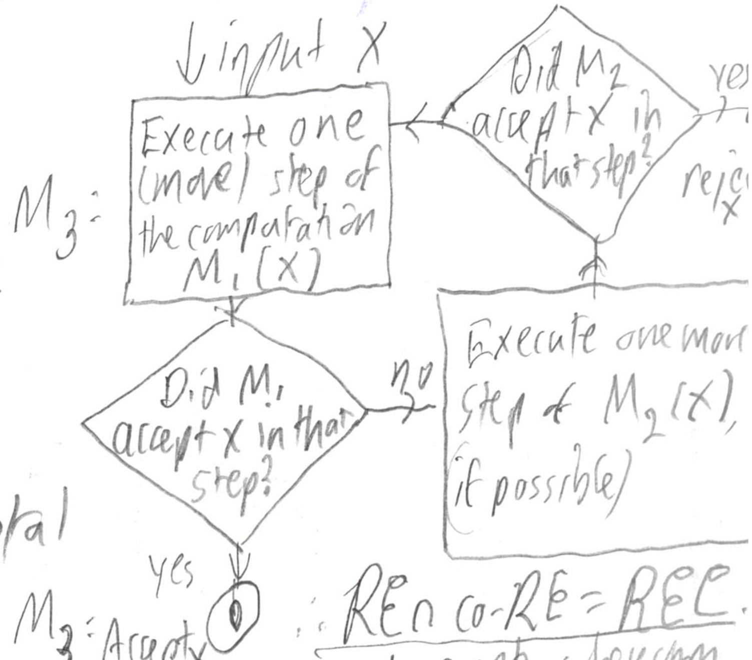
$L = \{x : x \text{ ends in } 0\}$. but swapping $F = \{s\}$ does not do $\tilde{L}$.

$\tilde{L} = \{x : x = \varepsilon \text{ or } x \text{ ends in } 1\}$

So does not show that $NP = co\text{-}NP$. Also does not show <u>NL = co-NL</u>, but this is a famous theorem (1988).

- Does not work if M does not halt for all inputs.

Proof: Take DTMs M_1 and M_2 such that $L(M_1) = A$ and $L(M_2) = \tilde{A}$. Build M_3 "via flowchart" as follows.



But we can show

<u>Theorem</u>: if A and $\tilde{A}$ are both c.e., then A is decidable.

Then M_3 is total because for all x, exactly one of $M_1(x)$ and $M_2(x)$ halts and accepts, whereupon $M_3(x)$ halts and decides. So A is decidable. ☒

$\therefore$ <u>$RE \cap co\text{-}RE = REC$</u>.

Theorem: We can define a language D in co-RE $\setminus$ RE,
so those classes are different and neither equals REC.

Proof: Recall every DTM M has a unique(?) string code $\langle M \rangle$.

Define $D_{TM} = \{\langle M \rangle : M \text{ does not accept } \langle M \rangle\}$.

Suppose D were c.e. Then there would be a DTM Q s.t.
$L(Q) = D$.

But then Q accepts $\langle Q \rangle \Leftrightarrow \langle Q \rangle \in D$ by $L(Q) = D$

$\Leftrightarrow \langle Q \rangle \notin D$ by def^n of D

$\therefore \langle Q \rangle \in D \Leftrightarrow \neg \langle Q \rangle \in D$.

A logical stmt can never be equivalent to its negation, else the Universe explodes. So D cannot be c.e.

Analogy: Any 1-1 function $f: A \to \mathcal{P}(A)$ cannot be onto $\mathcal{P}(A)$. $D_f = \{x : x \text{ is not in the set } f(x)\}$.
If D is in the range of f, then it would equal $f(q)$ for some $q \in A$. But then

$q \in D \Leftrightarrow q$ is in the set $f(q)$ by $f(q) = D$

$\Leftrightarrow q$ is not in the set $f(q)$, by def^n of $q \in D$.

Same contradiction. Hence $\mathcal{P}(\Sigma^*)$ is uncountable.

But D __is__ __co-r.e.__: The complement of D is (essentially)

$$K_{TM} = \{\langle M\rangle : M \text{ \underline{does} accept } \langle M\rangle.\}.$$

K_{TM} __is r.e.__: It is a "central slice" of the A_{TM} language

$$A_{TM} = \{\langle M, x\rangle : M \text{ accepts } x\},$$

and the UTM M_U s.t. $L(M_U) = A_{TM}$ can be modified to accept K_{TM} with an initial check that $x = \langle M\rangle$. ☒

Loose end: what if a given input x is not the code of any TM M? Several tech. answers (edit…

- Consider such x to be a code for $M_0 =$ … D … goes … ga…
- Use Gödel numbers:

$$x = \text{number } i_x \rightarrow \text{machine } M_{i_x}.$$

Historical Note: The latter gives us a computable enumeration $M_1, M_2, M_3, \ldots, M_i, \ldots$ of machines. We could build a "TM compiler" that would not only test whether a TM code is valid, it could list out all the valid codes ad-infinitum. __Any__ Turing machine can be considered to generate such a list, by making it try to generate all its own accepting computations. Then it __enumerates__ its language, hence the term "r.e."