

**Def<sup>n</sup>:** A language  $L$  is  $\begin{cases} \text{co-r.e.} \\ \text{co-c.e.} \end{cases}$  if its complement  $\tilde{L} = \Sigma^* - L$  is r.e. c.e.

**"Cone Diagram" for classes:**

**Complexity Theory Analogy**

**Convention 1:** Classes include regions underneath (unless otherwise indicated)

**Convention 2:** Left-Right Symmetry means the class  $C$  is closed under complements:  $L \in C \Leftrightarrow \tilde{L} \in C$

**Theorem:**  $RE \cap \text{co-RE} = \text{REC}$ .

Unpacked, this says that for any language  $L$ , if  $L$  is r.e. and  $L$  is co-r.e., then  $L$  is decidable, and vice-versa.

There is a TM  $M$  such that  $L(M) = L$  and there is a TM  $M'$  such that  $L(M') = \tilde{L}$ .

To prove the  $\Rightarrow$  part, our goal is: Show that  $L$  is decidable, by building a program  $P$  s.t.  $L(P) = L$  and  $P$  halts for all inputs (even if  $M$  and  $M'$  do not)  $\uparrow$

So we may take  $M$  and  $M'$  and build  $P$  as follows (as a flowchart diagram):

```

    graph TD
      Start(( )) --> Step1[Do one (more) step of the sim. M(x) (if possible)]
      Step1 --> Dec1{Did M(x) accept in that last step?}
      Dec1 -- yes --> Accept((Accept x))
      Dec1 -- no --> Step2[Do one (more) step of the simulation M'(x) (if possible)]
      Step2 --> Dec2{Did it accept in that last step?}
      Dec2 -- yes --> Reject((Reject x))
      Dec2 -- no --> Step1
  
```

So  $P$  is total and  $L(P) = L$ , which means  $L$  is decidable.  $\square$   $P(x)$  rejects when it sees that  $M'$  accepts.

**First Exam**  
Monday Oct 21?  
Tomorrow (Oct 1) 2pm  
Can be recitation section (default Davis 203 Theory Lab)  
Oct 8: make up lecture for 11th?

**Flowchart:**

```

    graph TD
      Start(( )) --> Step1[Do one (more) step of the sim. M(x) (if possible)]
      Step1 --> Dec1{Did M(x) accept in that last step?}
      Dec1 -- yes --> Accept((Accept x))
      Dec1 -- no --> Step2[Do one (more) step of the simulation M'(x) (if possible)]
      Step2 --> Dec2{Did it accept in that last step?}
      Dec2 -- yes --> Reject((Reject x))
      Dec2 -- no --> Step1
  
```

**Text:**

$P(x)$  always halts because for all  $x \in \Sigma^*$ , either  $x \in L$  so  $M(x)$  eventually halts and accepts, so  $P(x)$  accepts or  $x \in \tilde{L}$  so  $M'(x)$  eventually accepts — even if  $M$  halts and rejects  $x$  beforehand,  $P$  can just keep going, until  $P(x)$  rejects when it sees that  $M'$  accepts.

The lines cut off at the top said "input  $x \in \Sigma^*$ " and "yes  $\rightarrow$  Reject  $x$ ."

Def<sup>n</sup>: A language  $L \subseteq \Sigma^*$

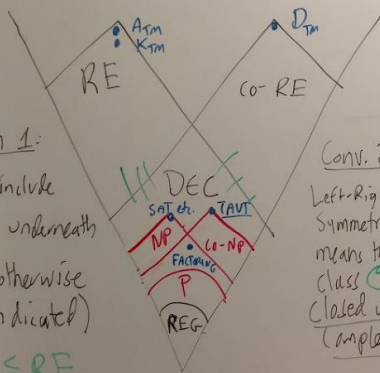
"Cone Diagram" for classes:

Complexity Theory Analogy

Convention 1:

Classes include regions underneath (unless otherwise indicated)

$DEC \leq RE$   
 $DEC \leq co-RE$



Key Definition: A language  $A \xrightarrow{\text{many-one reduces}} B$   $\xrightarrow{\text{mapping-reduces}}$  to a language  $B$ , if there is a **polynomial-time** computable function  $f: \Sigma^* \rightarrow \Sigma^*$  s.t.

$x \in A \iff f(x) \in B$ .

$x \in A \iff f(x) \in B$

Simple example: recall  $x \in K_{TM} \iff \langle x, x \rangle \in A_{TM}$

ie.  $\langle M \rangle \in K_{TM} \iff \langle M, \langle M \rangle \rangle \in A_{TM}$

ie.  $M$  accepts its own code  $\iff$  the universal  $M$  running  $M$

Conv. 2: Conv. 3: Left-Right Symmetry means the class  $C$  is closed under complements:  $L \in C \iff \bar{L} \in C$

Now notice that the function  $f(x) = \langle x, x \rangle$  is computable. (whatever convention we use)

So  $K_{TM} \leq_m^P A_{TM}$  via this  $f$ , which is simply in  $O(n)$  time just by copying the argument.

note:  $\bar{K}_{TM} = D_{TM}$  is not co, so  $RE$  is not closed under  $\bar{\cdot}$ .

$\xrightarrow{\text{many-one reduces}} B$   $\xrightarrow{\text{mapping-reduces}}$  to a language  $B$ , written  $A \leq_m^P B$

computable function  $f: \Sigma^* \rightarrow \Sigma^*$  s.t. for all  $x \in \Sigma^*$ ,

$f(x) \in B$

ie. recall  $x \in K_{TM} \iff \langle x, x \rangle \in A_{TM}$   
ie.  $\langle M \rangle \in K_{TM} \iff \langle M, \langle M \rangle \rangle \in A_{TM}$   
ie.  $M$  accepts its own code  $\iff$  the universal TM accepts when

running  $M$  on  $\langle M \rangle$ .  
whatever convention we use for pairing strings

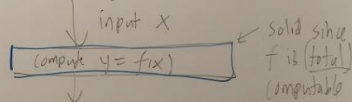
$A \leq_m^P A_{TM}$  via this  $f$ , which is simply computable in  $O(n)$  time just by copying the argument  $x$ .

Theorem (as implied by the diagram): If  $A \leq_m B$  then

- ① if  $B$  is c.e. then  $A$  is c.e.
- ② if  $B$  is decidable then  $A$  is decidable.
- ③ if  $B$  is co-c.e. then  $A$  is co-c.e.

Proof: First, for ①, let an  $M_B$  accepting  $B$  be given. Build

$M_A$  as follows.



Accept  $\rightarrow M_A$

$L(M_A) = A$  because

$M_A$  accepts  $x \iff M_B$  accepts  $f(x)$

$\iff f(x) \in B$  by def of the reduction.

So  $A$  is c.e.  $\iff x \in A$

Run  $M_B$  on  $y$  depth-endedly. If and when it accepts  $y$ .

For ②, if  $B$  is decidable then  $M_B$  can be total, so this becomes a solid box, so  $M_A$  is total too. ③ exercise.