CSE491/596 Lecture Mon. Oct. 19: Hardness, Completeness, and Degrees of Unsolvability

We have seen problem instances of type "A machine and a string" and "Just a machine". Another type is: "Two machines". For example, the language and problem EQ_{TM} is defined by:

Instance: Two deterministic Turing machines M_1 and M_2.
Question: Is $L(M_1) = L(M_2)$?

Show that the language of this problem is neither c.e. not co-c.e. Rather than do reductions from K and D, we can show this for a simplified special case: Fix M_2 to be some TM M_{all} whose language is Σ^*. Then the reduction $f(M) = \langle M, M_{all} \rangle$ has the property that

$$M \in ALL_{TM} \iff f(M) \in EQ_{TM}.$$

So it mapping-reduces ALL_{TM} to EQ_{TM}. Since we know that ALL_{TM} is neither c.e. nor co-c.e., the same is true of EQ_{TM}.

Note that the reduction "restricts" the second value to be a fixed machine. It is called a *restriction* of the more-general problem "to" a special case that reduces to it. Another example icomes from the following consequence of A_{TM} being the language of a universal Turing machine.

Theorem: For every c.e. language A, $A \leq_m A_{TM}$.

Proof. By A being c.e., there exists a DTM M_a such that $L(M_a) = A$. For the reduction, we need to define a function f of type "string" $\rightarrow$ "machine and a string" such that for all $x \in \Sigma^*$, $x \in A \iff f(x) \in A_{TM}$. So define $f(x) = \langle M_a, x \rangle$, or $f(x) = \langle a, x \rangle$ if you treat a as a Gödel number for the machine. Either way, the point is that a is fixed, so this is a linear-time computable reduction that just copies x onto other stuff. The reduction is correct because

$$x \in A \iff M_a \textit{ accepts } x \iff \langle M_a, x \rangle \in A_{TM} \text{ . } \boxtimes$$

Definition: Given any class $\mathbb{C}$ of languages, a language B is **hard** for $\mathbb{C}$ (**under** mapping reducibility $\leq_m$, which is the default---later poly-time mapping reductions $\leq_m^p$ will be the default) if for all languages $A \in \mathbb{C}$, $A \leq_m B$. If also $B \in \mathbb{C}$, then B is **complete** for $\mathbb{C}$ (under $\leq_m$), also called $\mathbb{C}$-**complete** when the reduicibility relation involved is understood.

Example: A_{TM} is hard for **RE** under $\leq_m$, and since A_{TM} is c.e., it is complete for **RE**.(also written **RE**-*complete* or *r.e.-complete*, less frequently "c.e.-complete."

Proposition:

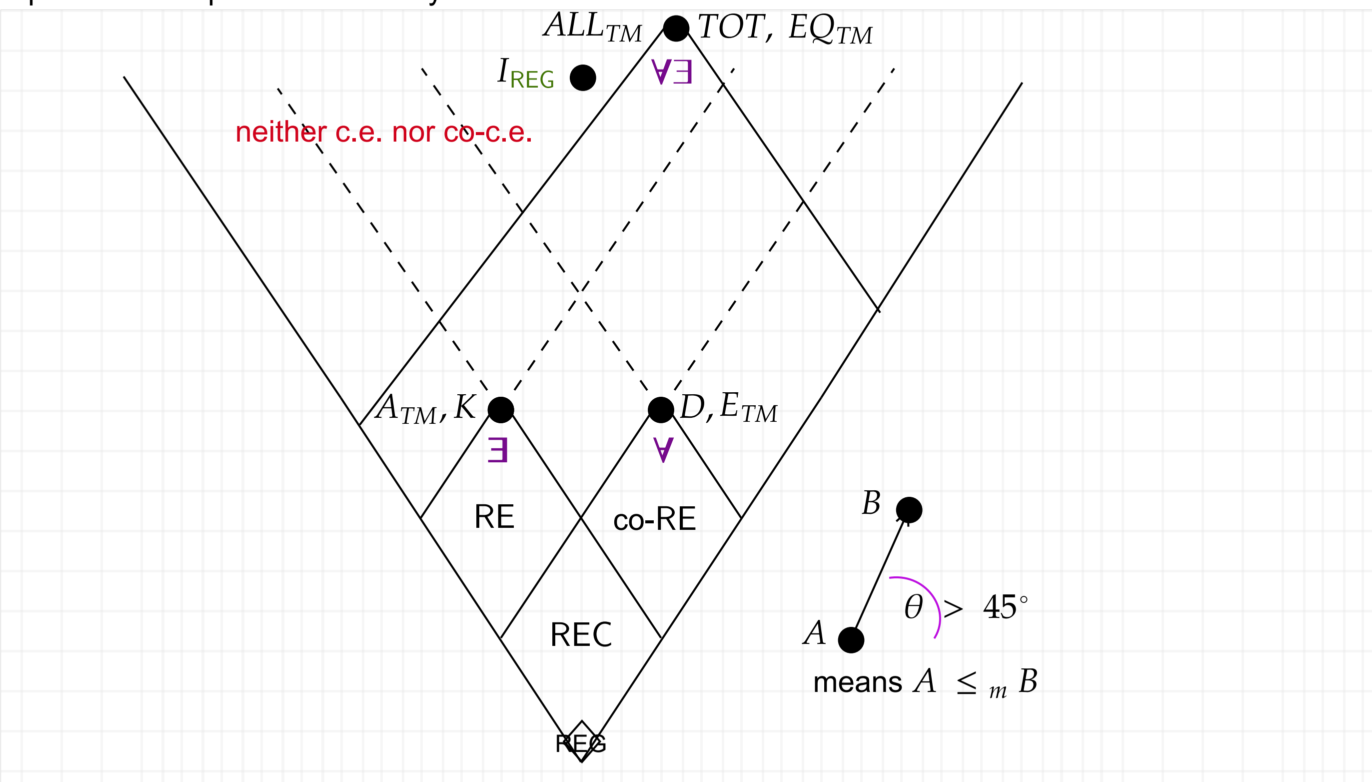
- If B is $\mathbb{C}$-hard and $B \leq_m E$, then E is $\mathbb{C}$-hard.
- If B is $\mathbb{C}$-complete, $B \leq_m E$, and E is in $\mathbb{C}$, then E is also $\mathbb{C}$-complete.
- If B and E are $\mathbb{C}$-complete (for any class $\mathbb{C}$), then $B \equiv_m E$. $\boxtimes$

Examples:

- NE_{TM}, HP_{TM}, and K_{TM} are also **RE**-complete just like A_{TM}.
- Likewise (by "mirror image"), D_{TM} and E_{TM} are complete for **co-RE**.
- ALL_{TM}, TOT, and EQ_{TM} are hard for both **RE** and **co-RE**, but are not complete for either class because they don't belong to either class. (They are in fact all complete for a higher-up class which is in a presentation topic; they are $\equiv_m$ equivalent to each other.)

Human Psych Fact #1: It is hard to think of an undecidable c.e. set that is **not RE**-complete. Giving one would be a fairly difficult homework problem---and if we considered completeness under the wider notion of *Turing reductions*, this was an open problem for two decades!

This anyway explains why I have been graphing these problems at the very peaks of classes---this represents completeness visually.



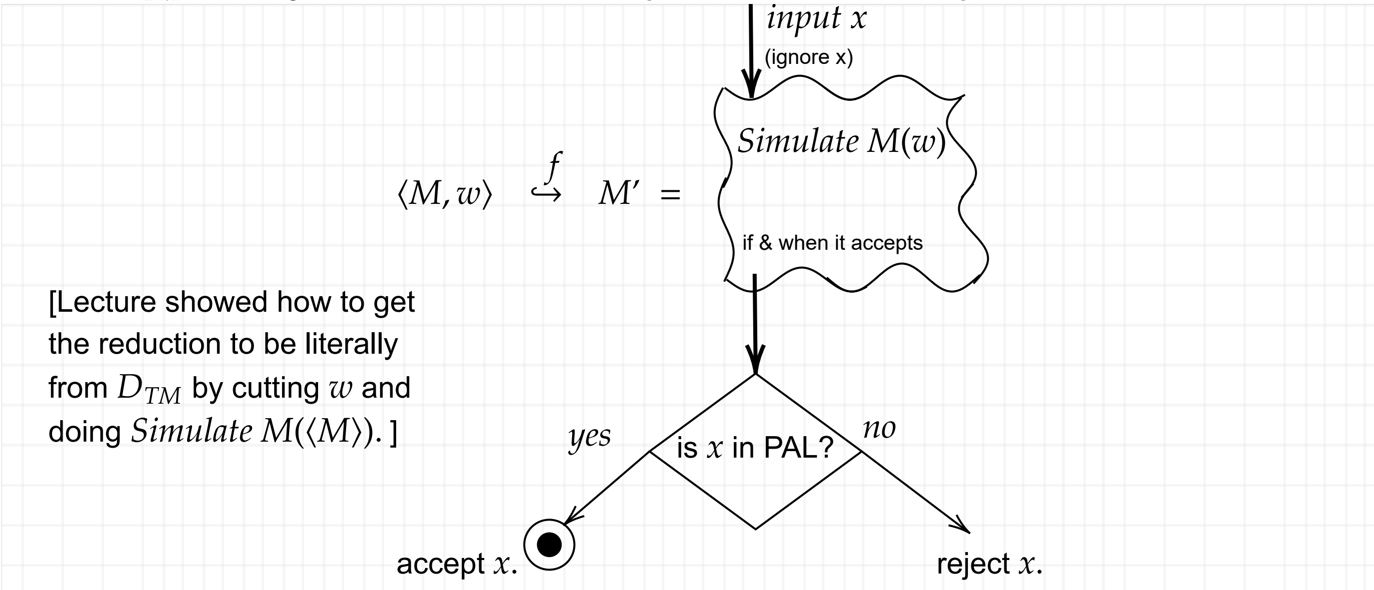
There are languages even higher up in the diagram. One of them---but we will only show that it is neither c.e. nor co-c.e.---is $\{\langle M \rangle : L(M) \text{ is regular}\}$. Sources call this REGULAR or $REGULAR_{TM}$ or REG_{TM}, but I will use the notation I_{REG} to signify that it is the *index set* of the class of regular languages.

Definition: For any class $\mathcal{C}$ of c.e. langauges (that is, $\mathcal{C} \subseteq$ RE), its **index set** is given by

$$I_{\mathcal{C}} = \{i : L(M_i) \in \mathcal{C}\}.$$

The traditional use of Gödel numbers here is why it is called an "index" set. The other way to think about it is that it is a "purely semantic property"---that is, a property of the language that a program accepts (or: of the Boolean function it computes) rather than of how the program is coded. For example E_{TM} is the index set of $\{\varnothing\}$, whereas $\varnothing$ is the index set of the *empty class*.

To show $?_{TM} \;\le\; I_{\text{REG}}$, we use the "all-or-nothing switch" as the first stage in a "filter":

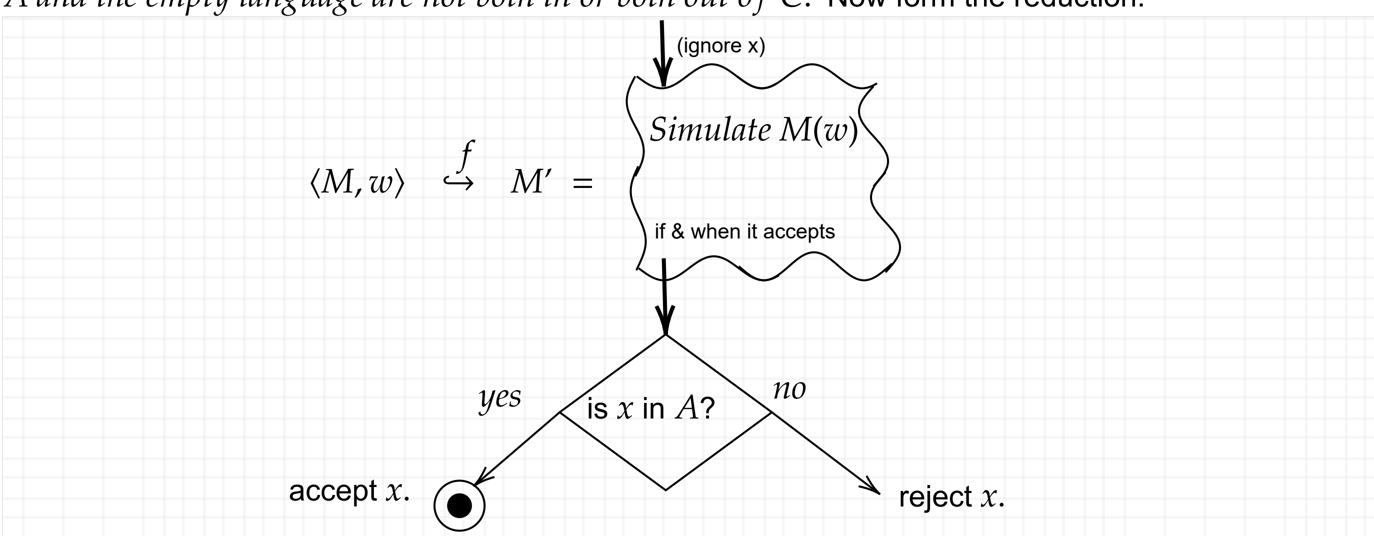


$\langle M, w\rangle \in A_{TM} \implies L(M') = PAL \implies L(M')\ \textit{is not regular} \implies \langle M'\rangle \notin I_{\text{REG}}$.
$\langle M, w\rangle \notin A_{TM} \implies L(M') = \varnothing \implies L(M')\ \textit{is regular} \implies \langle M'\rangle \in I_{\text{REG}}$.

So did we show $A_{TM} \le I_{\text{REG}}$? *No---the opposite*. We've reduced the complement of A_{TM} to I_{REG} . Knowing what we know about $\equiv_m$ now, we can tweak this to show $D_{TM} \le I_{\text{REG}}$. We have thus shown that I_{REG} is not c.e., besides being undecidable. [Showing that I_{REG} is not co-c.e. either is a self-study exercise. In fact, it is hard for ALL_{TM} but not equivalent to it---that gets difficult to show.]

Rice's Theorem: The only decidable index sets are $I_{\varnothing} = \varnothing$ and $I_{\text{RE}} = \mathbb{N}\ (\sim\, = \Sigma^*)$.

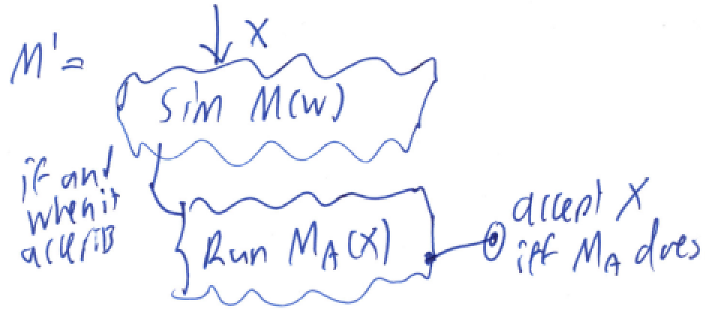
Proof: Because C is neither $\varnothing$ nor RE, we have a language A such that
A and the empty language are not both in or both out of C. Now form the reduction:



Proof: Given $I_{\mathcal{C}}$ where $\mathcal{C} \neq \emptyset$ and $\mathcal{C} \neq RE$, so there is a c.e. language A such that either: (a) $\emptyset \in \mathcal{C}$, $A \notin \mathcal{C}$ (*)

Take an M_A s.t. $L(M_A) = A$. or (b) $A \in \mathcal{C}$, $\emptyset \notin \mathcal{C}$.

Build $\langle M, w\rangle \xrightarrow{f} M' =$



In case (a),

$\langle M, w\rangle \in A_{TM} \Rightarrow L(M') = A \Rightarrow M' \notin I_{\mathcal{C}}$ $\therefore A_{TM} \leq_m \tilde{I}_{\mathcal{C}}$.

$\langle M, w\rangle \notin A_{TM} \Rightarrow L(M') = \emptyset \Rightarrow M' \in I_{\mathcal{C}}$

In case (b)

$\langle M, w\rangle \in A_{TM} \Rightarrow L(M') = A \Rightarrow L(M') \in \mathcal{C} \Rightarrow M' \in I_{\mathcal{C}}$ $\therefore A_{TM} \leq_m I_{\mathcal{C}}$

$\notin A_{TM} \Rightarrow L(M') = \emptyset \Rightarrow M' \notin I_{\mathcal{C}}$ since $\emptyset \notin \mathcal{C}$ in this case

Either way, $I_{\mathcal{C}}$ is undecidable. ☒

Example: $\mathcal{C} = REG$, case (b) applies with $A = \{$palindromes$\}$, so I_{REG} is undecidable.

Added:

(*) I could have worded this as, "Given any class $\mathcal{C}$, first suppose (a) that the empty language $\emptyset$ is in $\mathcal{C}$. E.g., $\mathcal{C}$ is the class of regular languages.

FYI, **Human Psych Fact #2** is that very few appealing concepts of language classes have definitions more complicated than ∃∀∃ in form. I_{REG} has that form because:

$$L(M) \text{ is regular} \iff \exists \text{ a DFA } M' \text{ such that } \langle M, M'\rangle \in EQ_{TM},$$

and the definition of $\langle M, M'\rangle \in EQ_{TM}$ has a (somewhat laborious) ∀∃ form. The index sets of all complexity classes we will study have similar form. Now on to Complexity Theory from Wed....