How to prove Turing decidability of languages

Language hierarchy

Recognizability

Reduce to $A_{TM}$: $R$ is T-recog if it is reducible to $A_{TM}$ ($R \leq_m A_{TM}$)

Reduce to recognizable language: If $R \leq_m R'$ and $R'$ is T-recog, then $R$ is T-recog

Give enumerator: $R$ is T-recog $\iff$ $\exists$ an enumerator $E$ such that $L(E) = R$

Give recognizer: $R$ is T-recog $\iff$ $\exists$ a Turing machine $T$ such that $L(T) = R$ (by existence of TM recognizer)

- WARNING: Be careful when saying stuff like “I can recognize if this happens by simulating $M$ on all inputs and checking if it accepts”. If you do something like this for, let’s say, $\overline{E}_{TM}$, then you have to make sure you use dovetailing so as to not get stuck in an infinite loop on a particular input.

$R$ is T-recog $\iff$ $\exists$ a language $D$, such that $R = \{ x \mid \exists y ((x, y) \in D) \}$ (by projection of decidable language)

Decidability

Give lexicographic-order enumerator: $A$ is T-decidable $\iff$ $\exists$ an enumerator $E$ such that $E$ prints all of the strings in $A$ in lexicographic order.

Show language and its complement are both recognizable: If $A$ is T-recog and $\overline{A}$ is T-recog then $A$ is T-decidable.

- I can take both recognizers and run them in parallel, simulating a step on each one, eventually one will accept, allowing me to decide $A$. It is important that you run them step-by-step in parallel, as opposed to first running the $A$ recognizer and then running the $\overline{A}$ recognizer. What if the first recognizer never halts?

Reduce to decidable language: If $D \leq_m D'$ and $D'$ is decidable, then $D$ is T-decidable (by mapping-reducibility to decidable language)

- Because I can map $D$ to $D'$, solve the $D'$ instance, and I will have solved the $D$ instance.

Undecidability

Reduce from $A_{TM}$: $U$ is undecidable if $A_{TM}$ is reducible to $U$ (by reduction from $A_{TM}$)

Reduce from undecidable problem: If $U' \leq_m U$ and $U'$ is undecidable, then $U$ is undecidable (by mapping-reducibility from undecidable language)
Turing-unrecognizability

If $A \leq_m B$ and $A$ is not T-recognizable, then $B$ is not Turing-recognizable (by mapping-reducibility to unrecognizable language).

If $A$ is not decidable, then $A$ or $\overline{A}$ is not Turing-recognizable.

If $J$ is undecidable and $J \leq_m \overline{J}$, then both $J$ and $\overline{J}$ are not Turing-recognizable.

Examples

Decidable: $A_{DFA}, E_{DFA}, EQ_{DFA}, A_{CFG}, E_{PDA}, A_{LBA}$

Undecidable: $A_{TM}, HALT_{TM}, ALL_{PDA}, EQ_{CFG}, E_{LBA}, PCP$. Also $ALL_{TM}$.

Unrecognizable: $\overline{A_{TM}}, E_{TM}, EQ_{TM}, EQ_{TM}$