Homework 10: The Asymmetry of Recognizability

CSCI 161

Due 9:00pm, December 3, 2020

Ben Dreyer

Overview. This last homework focuses on the differences between decidability and recognizability including questions about mapping reductions. Submit your edited copy of this text file with its compiled pdf for 5% extra credit. You may submit late only in accordance with the syllabus policy of 10% deduction per hour, but only up until 2:15 on the last day of classes (no extensions!) so we can discuss solutions for this and any other homework on our last review day.

Question 0. Who did you collaborate with on this homework? Optionally if you are willing to have me share your advice for students who take this course in the future, please let me know what was most and least helpful about studying with other students for this class this quarter.

Response. Talked with students in class about number 1. Textbook.

Question 1. (6pts) Recall that an LBA is a Turing machine whose tape head never leaves the portion of the tape initially occupied by the input, and let $EQ_{\text{LBA}} = \{\langle M_1, M_2\rangle \mid M_1, M_2$ are LBAs and $L(M_1) = L(M_2)\}$. Prove whether EQ_{LBA} is decidable, whether it is Turing-recognizable, and whether it is co-Turing-recognizable.

Solution. We can show that EQ_{LBA} is undecidable because E_{LBA} which we know is undecidable can be reduced to it. To do that, Assume D decides EQ_{LBA} and construct a Turing Machine S as such:

S = on $< M >$ M is a TM

1. Run D on $< M, M_{emptyset} >$
2. If D accepts: accept

   else: reject

We can see here that we have a mapping reduction from E_{LBA} to EQ_{LBA} because we havea computable function $f(M) = f(M, M_{emptyset})$ ($E_{\text{LBA}} \leq_m EQ_{\text{LBA}}$. Therefore we can deduct the EQ_{LBA} is undecidable since E_{LBA} is undecidable as well. We also know that E_{LBA} is Corecognizable and not recognizable, so we can say the EQ_{LBA} is also not recognizable due to inheritance.
To prove that EQ_{LBA} is co-recognizable we can construct a recognizer S that recognizes $\overline{EQ_{\text{LBA}}}$ as follows:

1. Run S on input $< M1, M2 >$
2. Run the following on strings of increasing size for an increasing number of steps:
   (a) If the string is accepted by exactly one of M1 and M2, accept

Question 2. (4pts) Recall that A_{TM} reduces to $REGULAR_{\text{TM}} = \{\langle M\rangle \mid M$ is a TM and $L(M)$ is regular$\}$. Prove whether $REGULAR_{\text{TM}}$ is Turing-recognizable and whether it is co-Turing-recognizable.

Solution. Since we know that $\overline{A_{\text{TM}}} \leq_m REGULAR_{\text{TM}}$, and we know that $\overline{A_{TM}}$ is not recognizable, we can also say the $REGULAR_{\text{TM}}$ is not Turing Recognizable.
To prove that $REGULAR_{TM}$ is co-Turing-recognizable we can construct the following machine that co-recognizes it:

S = on input $< M >$ M is a TM

1. Generate strings of increasing sizes from regular languages (finite) for an increasing but finite number of steps
2. For each string w generated:
3. Simulate M on the current string w
4. If M rejects w: reject

Question 3. (4pts) Let $J = \{w \mid w = 0x \text{ for some } x \in A_{\text{TM}} \text{ or } w = 1y \text{ for some } y \in \overline{A_{\text{TM}}}\}$. Prove that J is neither Turing-recognizable nor co-Turing-recognizable. For 2pts extra credit, also show $J \leq_m \overline{J}$.

Solution To prove this, use 2 arguments, one for not corecognizable, on for not recognizable.

Proof 1: Not corecognizable:

Assume $f(x) = 0x$, if $x \in A_{TM}$, then $x \in J$, likewise, if $x \notin A_{TM}$ then $x \notin J$. This means we have a computable function, and therefore a mapping reduction $A_{\text{TM}\leq_m} J$. This means J is not corecognizable because A_{TM} is not corecognizable.

Proof 2: Not recognizable:

Assume $f(x) = 1y$, if $y \in \overline{A_{TM}}$, then $x \in J$, likewise if $y \notin \overline{A_{TM}}$ then $x \notin J$. So we have $x \in J$ iff $y \in \overline{A_{TM}}$. So we have a computable function and therefore, a mapping reduction $\overline{A_{\text{TM}}} \leq_m J$. Becuase of our mapping reduction and that we know $\overline{A_{TM}}$ is not recognizable, we can say that J is also not recognizable.

Question 4. (4pts) Prove that a language A is Turing-recognizable if and only if $A \leq_m A_{\text{TM}}$.

Structuring your proof: Remember that a good strategy to prove an if and only if statement is to give two arguments for the two implications. Assuming A is recognizable, show there must be a mapping reduction. Assuming there is a mapping reduction, show that A must be recognizable. Use the definitions of recognizability and mapping reductions!

Solution. Argument 1
Assume we already know for some language A that $A \leq_m A_{\text{TM}}$, because Atm is recognizable, and A mapping reduces to it, A is also recognizable via inheritance.

Argument 2

Suppose we don't already know $A \leq_m A_{\text{TM}}$ for some language A. Assume A is recognizable, this means that some Turing Machine M, accepts it. From this logic, we can say that if x is an element of A, then M,x is an element of Atm because M is the Turing machine that accepts x. This means we have a mapping reduction using the function $f(x) =< M, x >$ and our mapping reduction is $A \leq_m A_{\text{TM}}$. Therefore, every recognizable language A must mapping reduce to Atm.

Brain teaser. Is there a nonregular language that can be map-reduced to a regular language? Can every nonregular language be map-reduced to a regular language?