COMP170 Recitation 5

Prove that \( HOMETOWN \leq_T HALT \)

We will show that \( HOMETOWN \) is decidable using a machine with an oracle to \( HALT \) as follows:

\[ M^{HALT} \text{ on input } \langle F, B \rangle \]

Step 1 Ask the oracle \( \langle F, "ben" \rangle \in HALT \)

Step 1A If yes, Run \( F \) on “ben”
   if it accepts, REJECT
   if it rejects, continue to step 2.

Step 1B If no, continue to step 2

Step 2 Ask the oracle \( \langle B, "ben" \rangle \in HALT \)

Step 2A If yes, Run \( B \) on “ben”
   if it accepts, ACCEPT
   if it rejects, REJECT

Step 2B If no, REJECT

Claim \( M^{HALT} \) is a decider for \( HOMETOWN \).
Consider the case when \( \langle F, B \rangle \) is in \( HOMETOWN \). The we know \( F \) did not accept “ben” and \( B \) did accept “ben”. Since \( F \) did not accept “ben” we know that it either rejected “ben” or looped on input “ben”. If it looped then the oracle said no and the machine moved to step 2. If it rejected, the machine halts, so the oracle said yes and ran \( F \) on input “ben”. \( F \) will reject so \( M^{HALT} \) will continue to step 2. If \( B \) accepts “ben” then the oracle will say yes and \( M^{HALT} \) will simulate \( B \) on input “ben”. Since we know \( B \) accepts then \( M^{HALT} \) accepts as it should.

Consider the case when \( \langle F, B \rangle \) is not in \( HOMETOWN \). Then we know either \( F \) accepted “ben” or \( B \) did not accept “ben”. Consider the case when \( F \) accepted “ben”. We know that the oracle will answer yes and \( M^{HALT} \) will simulate \( F \) on input “ben” and it will accept, so \( M^{HALT} \) rejects as it should. If \( B \) did not accept “ben” (and \( F \) did not accept “ben”) then we know that either \( B \) rejects “ben” or \( B \) loops on “ben”. If \( B \) loops on “ben” then the oracle said no in step 2 and \( M^{HALT} \) rejected as it should. If \( B \) rejects “ben” then the oracle says yes and then \( M^{HALT} \) simulates \( B \) on “ben” and since \( B \) rejects “ben” then \( M^{HALT} \) rejects as it should.

REVIEW

Theorem: \( EMPTY \leq_T A_{TM} \)

Proof Idea: To show that a set, \( A \), Turing reduces to another set \( B \), we need to show that \( A \) can be decided using a Turing machine with oracle access to \( B \). To show that \( EMPTY \) Turing reduces to \( A_{TM} \) we have to show that \( EMPTY \) can be decided by a Turing machine with oracle access to \( A_{TM} \). To do this we will use the recognizer for \( EMPTY \). If a recognizer for \( EMPTY \) accepts
we know that the input machine, $M$, accepted some input. We will create a new machine $M'$ that accepts all of its inputs if the recognizer for $\text{EMPTY}$ accepts the input machine $M$. We then ask the oracle to $A_{TM}$ what $M'$ does on a fixed input. If it accepts, then we reject. If it rejects, we accept.

**Lemma:** $\text{EMPTY}$ is recognizable

**Proof of Lemma:**
Consider the machine $R_E$ defined below.

$$R_E$$ on input $\langle M \rangle$

for $i : 0$ to $\infty$

Run $M$ on $s_0, s_1, \ldots s_i$ for $i$ steps

If $M$ accepts some input $s_j$, ACCEPT

$R_E$ is a valid Turing Machine, as we only simulate another valid Turing Machine for a finite number of steps and witness its behavior over those steps. If $M$ accepts any input, $R_E$ will eventually witness that acceptance and accept the machine $M$. That is, if $M \in \text{EMPTY}$, $R_E$ accepts, so it is a recognizer for $\text{EMPTY}$

**Proof of Theorem:**
Let $R_E$ be the recognizer for $\text{EMPTY}$. Now for every Turing Machine $M$ consider the corresponding machine $MR_M$ defined as follows

$MR_M$ on input $x$

Run $R_E$ on input $\langle M \rangle$

If it accepts, ACCEPT.

If $M$ is valid Turing Machine, so is $MR$ as we known the recognizer for $\text{EMPTY}$ exists and is well defined over the space of all Turing Machines.
Now consider the oracle machine $M^{A_{TM}}$ defined as follows:

$M^{A_{TM}}$ on input $\langle M \rangle$

- Query $\langle MR_M, 17 \rangle \in A_{TM}$ where $MR_M$ is the corresponding machine for input $M$ defined above.

  - If the oracle responds yes, REJECT
  - If the oracle responds no, ACCEPT.

Claim $M^{A_{TM}}$ decides $\emptyset$. It is a valid oracle Turing Machine as it queries a valid input to $A_{TM}$. Further, if $M$ is empty then the recognizer for $\overline{\emptyset}$ will never accept, thus $MR_M$ will never accept any input including 17. So the query to $A_{TM}$ will be answered no and the machine ACCEPTS. If $M \notin \emptyset$, then $R_E$ will recognize it and accept, and therefore $MR_M$ accepts everything. The query to the oracle will be answered yes, and the machine will reject. Q.E.D.

Notice that since this is a decider for $\emptyset$ it can be turned into a decider for $\overline{\emptyset}$ by flipping the accept and reject states. In fact, for all $A, B$ if $A \leq_T B$ then $\overline{A} \leq_T B$. Also, if $A \leq_T B$ then $A \leq_T \overline{B}$, here you just flip your yes and no answers on your query.