COMP170 Halting Problem

The famous Halting problem is defined as the set \( \text{HALT} = \{ \langle M, w \rangle \mid M \text{ halts when started on input } w \} \). Recall that when we say a machine halts on an input, it is the same as saying that the machine accepted or rejected that input. The complement of this set is the set of \( \langle M, w \rangle \) where \( M \) is a machine that does not halt on input \( w \). (Namely, \( M \) loops on \( w \).)

We will prove this set is undecidable by doing a proof by contradiction. The proof by contradiction will assume that \( \text{HALT} \) is decidable. We then use the decider for \( \text{HALT} \) to decide \( A_{TM} \) which we know to be undecidable, thus creating a contradiction.

**Theorem** \( \text{HALT} \) is undecidable

**Proof by contradiction.**

We proceed by contradiction, that is, assume that \( \text{HALT} \) is decidable. Since \( \text{HALT} \) is decidable we know there is a machine that decides it, let \( M_{HALT} \) be that machine. We will use this machine to create a decider for \( A_{TM}, M_{ATM} \).

\( M_{ATM} \) on input \( \langle M, w \rangle \)

1. Run \( M_{HALT} \) on \( \langle N_{\langle M,w \rangle}, 5 \rangle \) /* \( N_{\langle M,w \rangle} \) is defined below*/
   1a. If \( M_{HALT} \) accepts \( \langle N_{\langle M,w \rangle}, 5 \rangle \), ACCEPT.
   1b. If \( M_{HALT} \) rejects \( \langle N_{\langle M,w \rangle}, 5 \rangle \), REJECT.

Now consider the machine \( N_{\langle M,w \rangle} \) defined below. Note that it depends on the \( \langle M, w \rangle \) given as input. This is fine, we just need to know that it is a valid Turing machine. You can imagine it is predefined ahead for all possible \( M \) and \( w \).

\( N_{\langle M,w \rangle} \) on input \( x \)

1. if \( x == 5 \)
   1a. Run \( M \) on \( w \)
      i. If \( M \) accepts \( w \), ACCEPT
      ii. If \( M \) rejects \( w \), loop
   2. else loop

Claim, \( M_{ATM} \) decides \( A_{TM} \).

We can see that \( M_{ATM} \) is a valid Turing machine if \( M_{HALT} \) is a valid Turing machine, as it has a finite number states and only makes calls to \( M_{HALT} \). Since we have also assumed that \( M_{HALT} \) is a decider, then we know \( M_{ATM} \) halts on all inputs. Now we need only show that it is a decider for \( A_{TM} \). There are two cases two check, \( \langle M, w \rangle \in A_{TM} \) and \( \langle M, w \rangle \notin A_{TM} \).

Case \( \langle M, w \rangle \in A_{TM} \). Since \( \langle M, w \rangle \in A_{TM} \) we know that \( M \) accepts \( w \), then we know that the corresponding \( N_{\langle M,w \rangle} \) will accept 5, by definition. Further, we know since it accepted 5, it halted on 5 so that \( M_{HALT} \) accepted \( \langle N_{\langle M,w \rangle}, 5 \rangle \). This implies that \( M_{ATM} \) accepted, so it decided this case correctly.

1
Case $(M, w) \notin A_{TM}$. Since $(M, w) \notin A_{TM}$ there are two sub-cases to check, $M$ rejects $w$, and $M$ loops on $w$. Consider the case when $M$ rejects $w$. Then by definition, $N_{(M, w)}$ will loop on 5, and since $M_{HALT}$ is a decider it will reject $(N_{(M, w)}, 5)$ and similarly $M_{ATM}$ will reject. Therefore $M_{ATM}$ decides this sub-case correctly. Now consider the sub-case when $M$ loops on $w$. Then when $N_{(M, w)}$ is run on input 5 it simulates $M$ on $w$ and will loop during the simulation. This will in turn cause $M_{HALT}$ to reject $(N_{(M, w)}, 5)$, and $M_{ATM}$ will reject this $(M, w)$, correctly deciding this sub-case.

Since $M_{ATM}$ halts on every input and correctly decides all cases, it decides $A_{TM}$. This is a contradiction as no such machine can exist. Since the only assumption made in the creation of $M_{ATM}$ was that machine $M_{HALT}$ decides $HALT$. It must be that no such machine exits. Thereby showing that $HALT$ is undecidable.