ICCS310: Assignment 5

Natthakan Euaumpon
natthakaneuaumpon@gmail.com
March 2021

1: Reject TM

Let AFSOC that REJECT$_{TM}$ is decidable. This mean there is a Turing Machine which decides it. Let the machine $R(< M, x >)$.

$$R(< M, x >) = \begin{cases} \text{reject if M accept x,} \\ \text{accept if M reject x,} \\ \text{reject if M loop on x} \end{cases}$$

Using R we can create another machine D. Where D $< M >$ where M is a machine. To prove this we run R on $< M, < M >>$. From this we have:

$$D(< M >) = \begin{cases} \text{reject if M accept } < M >, \\ \text{accept if M reject } < M >, \\ \text{reject if M loop on } < M > \end{cases}$$

Applying to $M = D$ we have:

$$D(< D >) = \begin{cases} \text{reject if D accept } < D >, \\ \text{accept if D reject } < D >, \\ \text{reject if D loop on } < D > \end{cases}$$

This contradict, so D and R can't exist. Therefore, REJECT$_{TM}$ is undecidable.

2: Accept VS. Reject

---

(1)
Prove that ACCEPT$_{TM}$ $\leq$ REJECT$_{TM}$:
WLOG let assume that there are 2 machine that decides ACCEPT$_{TM}$ and REJECT$_{TM}$, M_a and M_r. Where M_a decides ACCEPT$_{TM}$ and M_r decides REJECT$_{TM}$ respectively. The given input is $< M, x >$ let:

1. Create $M^{'}$ from M by reversing accepting and rejecting states.
2. Run M_r with $< M^{'}, x >$
3. If M_r accept then we reject and viceversa

From this we know that M_a will be able to decide ACCEPT$_{TM}$ if it is given a REJECT$_{TM}$. Therefore, ACCEPT$_{TM}$ $\leq$ REJECT$_{TM}$.

(2)
Prove that REJECT$_{TM}$$\leq$ ACCEPT$_{TM}$:
WLOG let assume that there are 2 machine that decides ACCEPT$_{TM}$ and REJECT$_{TM}$, M_a and M_r. Where M_a decides ACCEPT$_{TM}$ and M_r decides REJECT$_{TM}$ respectively. The given input is $< M, x >$ let:

Natthakan Euaumpon
natthakaneuaumpon@gmail.com

1. Create $M^{'}$ from M by reversing accepting and rejecting states.
2. Run M_a with $< M^{'}, x >$
3. If M_a accept then we reject and viceversa

From this we know that M_r will be able to decide REJECT$_{TM}$ if it is given an ACCEPT$_{TM}$. Therefore, REJECT$_{TM}$ $\leq$ ACCEPT$_{TM}$.

3: Reverse on TM

AFSOC that T is deciable. Then let M_T be recogniser of T sp that we can construct a recogniser of A_{TM}. Let $M' =$ on input $< M, w >$. First, lets construct a TM $M_w =$ on input x.

$$M_w = \text{on input } x \begin{cases} \text{reject if } x = rev(w), \\ \text{run M on w and return the result if } x = w, \\ \text{reject if it's not fit on either of the above condition} \end{cases}$$

Then run M_T on $< M_w >$. From this we know that $M^{'}$ accept $< M, w >$ iff M_T accept $< M, w >$ iff M doesn't accept w iff $< M, w > \in \bar{A_{TM}}$. Therefore T is undeciable.

4: Undecidability

(1)
Show that TOTAL is undeciable:
AFSOC that TOTAL is decidable. This means there is a Turing Machine T that decides it. Then we can use $T(< M, x >)$ where x is an input String. Let define a new machine N. Where N run on input y. Then we run the machine.

1. If y is not the same as x hault
2. Feed y to machine M and let M compute x.
3. If M's computation on input x halts and rejects x, loop indefinitely. Else if M's computation on input x halts and accepts x, halt. Else continue looping.

Then feed the string $< N >$ into T. The machine will return output if T accepts $< M, x >$. Which is impossible, so TOTAL is undecidable.

(2)
Show that FINITE us undecidable:
AFSOC that FINITE is decideable. Let T be a TM that decide FINITE. Then we can construct another TM called U which will be use to decide A_{TM}. Let U = on input $< M, w >$ where w is an input string. Then we construct $< M_w >$ where:

$$M_w = \text{on input } x \begin{cases} \text{reject if M reject w,} \\ \text{accept if M accept w} \end{cases}$$

Run T on M_w. If T accept reject and viceversa. If M accept w then it accept all input which means it is indefinite language. Hence T reject M, therefore FINITE is undefineable.

(3)
Show that REGULAR is undecidable:
AFSOC that REGULAR is decidable. This mean there is a turing machine T that can decide it. Next, let construct another TM called U which will be use to decide A_{TM}. Let U = on input $< M, w >$ where w is an input string. Then we construct $< M_w >$ where:

$$M_w = \text{on input } x \begin{cases} \text{accept if w if in the form of } 0^1 n^1, \\ \text{reject otherwise} \end{cases}$$

From this we know that REGULAR is undecidable.

5: TOTAL Is No Harder Than Finite

WLOG let assume that there are 2 macine M_t and M_f where M_t decides TOTAL and M_f decides FINITE respectively. The given input is $< M, x >$ where x is an input string let:

1. Create $M^{'}$ from M by reversing accepting and rejecting states.
2. Run M_f with $< M^{'}, x >$
3. If M_f accept then we reject and viceversa

From this we know that M_t will be able to decide TOTAL if it is given a FINITE. Therefore, TOTAL $\leq$ FINITE.

6: FINITE Is No Harder Than TOTAL

WLOG let assume that there are 2 macine M_t and M_f where M_t decides TOTAL and M_f decides FINITE respectively. The given input is $< M, x >$ where x is an input string let:

1. Create $M^{'}$ from M by reversing accepting and rejecting states.
2. Run M_t with $< M^{'}, x >$
3. If M_t accept then we reject and viceversa

From this we know that M_f will be able to decide FINITE if it is given a TOTAL. Therefore, FINITE $\leq$ TOTAL.