Proving decidability of language

Consider the Turing machine *M* which is use for recognizing the language *A* in such a way that $A = A(M)$.

So, it can be said that language *A* is Turing-recognizable or even it can be said that it is recognizable.

A Turing machine is use for deciding the language *A* if $A = A(M)$ and Turing machine *M* hold for each and every input.

So, it can be said that *A* is decidable if and only if Turing machine *M* is use for deciding *A*.

Suppose, $A \le_m \overline{A}$, then it is quite obvious $\overline{A} \le_m A$ also exists by using the same mapping reducibility function.

A is Turing recognizable then $\overline{A}$ is also recognizable as follows:

Assume *M* is the recognizer for Turing Machine $\overline{A}$ and *N* is the recognizer for *A*. *F* is the reduction function for *A* to $\overline{A}$.

N can be described as:

N is the recognizer and recognizes input or string *w*.

N = Input *w*:

• Compute *F(w)*: *F(w)* function is mapping function that computes mapping reducibility between Turing Machines *P* and *Q*.

• Run *M* on input *F(w)* and output whatever *M* output.

As *M* is the recognizer for $\overline{A}$, now run the output *F(w)* on *M* to Find Mapping reducibility between Turing Machines *A* and $\overline{A}$.

This implies that $\overline{A}$ is also Turing Recognizable.

If A and $\overline{A}$ is Turing recognizable then it can be proved that $A \le_m \overline{A}$ is also decidable.

A language is decidable if its components are Recognizable or co-recognizable as it is already proved that *A* and $\overline{A}$ both are recognizable then consider *P* for deciding the language *A*.

Let P_A and $\overline{P_A}$ is use for deciding that A and $\overline{A}$ is recognizable.

• For any value of input *x* whether it is 1,2, 3 user need to simulate the value for P_A and $\overline{P_A}$ for the finite number of steps. If $x \in A$ then simulation is accepted and if there is the situation that $x \notin A$ then simulation is halted.

• Run both the decider P_A and $\overline{P_A}$ in parallel for the particular input *x* till either of them accepts.

• If $\overline{P_A}$ is accepted then accept it for the particular value of *x* and then halt. If $\overline{P_A}$ is accepted then reject the particular value of *x* and after that halt the Turing machine.

Running P_A and $\overline{P_A}$ in parallel means Turing Machine have 2 tapes 1 for simulating P_A and another for simulating $\overline{P_A}$ it continue until one of them accepts.

Now, it is quite obvious that input *x* is whether running on P_A or $\overline{P_A}$ so it must be accepted by one of them Turing Machine is halted whenever P_A or $\overline{P_A}$ accepts *x*. It accepts all strings in *A* and rejects all strings in *A* so P_A is decider for *A* and *A* is decidable.

As, for every input, Turing machine is halted for each and every input, then it can be said that $A \le_m \overline{A}$ is decidable.