L11: Church-Turing Thesis and Examples of Decidable Languages

3.17) Let $B = \{(M_1), (M_2), \ldots\}$ be a Turing-Recognizable language consisting of Turing Machine. Descriptions. Show that there is a decidable language C consisting of Turing Machine descriptions such that every machine described in B has an equivalent machine in C and vice versa.

Enumerator – An enumerator is a Turing machine that consist a work tape and the output tape. It output strings by using the work tape without accepting any input.

Theorem: "A language is Turing-Decidable if and only if some enumerator enumerates the strings of this language in lexicographic order.

Consider the language $B = \{(M_1), (M_2), \ldots\}$.

B is a Turing recognizable language.

C is a language consisting of Turing machine descriptions.

Consider $\mathcal{E}$ be the enumerator for the Turing recognizable language B.

Construct an enumerator $\mathcal{E}_0$ which output the strings of C in lexicographic order.

From the above theorem, C is decidable.

Enumerator $\mathcal{E}_0$ simulates $\mathcal{E}$.

When $\mathcal{E}$ gives the i^{th} TM (M_i) as output, then enumerator $\mathcal{E}_0$ pads M_i by adding sufficiently many extra useless states to obtain a new TM M_i' where the length of (M_i') is greater than the length of (M_{i-1}'). Then $\mathcal{E}$ outputs (M_i')

Thus simulation occurs in both directions.

Therefore, $\mathcal{E}_0$ and $\mathcal{E}$ are equivalent.