5CCS2FC2: Foundations of Computing II

Tutorial Sheet 2

2.1 Describe an algorithm, that can be implemented on a Turing Machine, that accepts the following language

$$\mathsf{E}_{DFA} = \{\langle A \rangle \ : \ A \text{ is a DFA such that } L(A) = \emptyset\}$$

In other words, E_{DFA} is the language of all DFAs that do not accept any words (including the empty string ε).

2.2 Show that the following language is decidable by reducing it to the language E_{DFA},

$$\mathsf{E}_{NFA} = \{\langle A \rangle \ : \ A \text{ is an NFA such that } L(A) = \emptyset\}$$

2.3 *(Tricky!)* Show that the following language is decidable, by reducing it to the language E_{DFA}:

$$\mathsf{EQ}_{DFA} = \{\langle A, B \rangle \ : \ A \text{ and } B \text{ are DFAs such that } L(A) = L(B)\}$$

In other words, EQ_{DFA} is the language of all pairs of *'equivalent'* DFAs that accept precisely the same words.

2.4 Show that the following language is undecidable

$$\mathsf{EQ}_{TM} \;=\; \{\langle M_1, M_2\rangle : M_1, M_2 \text{ are TMs such that } L(M_1) = L(M_2)\}.$$

by a reduction from the language E_{TM}.

2.5 (i) Show that the language A_{TM} is recursively enumerable by constructing a sound and complete algorithm that recognises all words $\langle M, w\rangle$, where M encodes a TM that accepts w.

(ii) Hence, or otherwise, show that its complement $\overline{\mathsf{A}_{TM}}$ is *not* recursively enumerable.

2.6 *(Tricky!)*

(i) Show that the language $\overline{\mathsf{EQ}_{TM}}$ is not recursively enumerable by reducing A_{TM} to its complement EQ_{TM}. (In other words, that EQ_{TM} is not co-recursively enumerable.)

(ii) Show that the language $\overline{\mathsf{EQ}_{TM}}$ is also not co-recursively enumerable by reducing A_{TM} to $\overline{\mathsf{EQ}_{TM}}$. (In other words, that EQ_{TM} is not recursively enumerable.)

(It follows that $\overline{\mathsf{EQ}_{TM}}$ and EQ_{TM} are 'harder' than any recursively enumerable or co-recursively enumerable problem. There are not even any sound-and-complete algorithms for either problem)