Date: 2021/05/03 Duration: 90 minutes

Read the instructions carefully.

1. Let Σ be a finite alphabet and let '$\leq$' be a total order on Σ. This order can be extended to a total order on Σ^* as follows. For $x, y \in \Sigma^*$, we say $x \leq y$ if one of the following is true.

   - $|x| < |y|$.
   - $|x| = |y|$ and y does not precede x in the dictionary order induced by '$\leq$'. (Formally, there doesn't exist an index i such that $x[1..i-1] = y[1..i-1]$ and $y[i] < x[i]$.)

   Recall the definition of an enumerator and an enumerable language from Homework 4. We say that a language L is *sorted-enumerable* if there exists an enumerator which prints all strings in L exactly once and in sorted order according to '$\leq$', and doesn't print any string not in L.

   1. [**3 marks**] Prove that every decidable language is sorted-enumerable.
   2. [**3 marks**] Prove that every sorted-enumerable language is decidable.

2. [**4 marks**] Fix a finite alphabet Σ and, as usual, for every $w \in \Sigma^*$, let M_w denote the Turing machine over Σ whose description is w (if w doesn't legally describe a Turing Machine, then M_w is some fixed Turing machine, say the one whose initial and reject states are the same, has no state other than the accept and the reject state, and has no extra tape symbol apart from the alphabet and the blank symbol). Let `InfVisit` be the set of all strings w for which there exists an $x \in \Sigma^*$ such that M_w when run on x visits every cell of its tape infinitely many times. Is `InfVisit` decidable? Prove your answer.