Date: 2021/05/16 Duration: 2 hours

Read the instructions carefully.

More instructions specific to this exam:

1. Answers must be accompanied by rigorous proofs. Unlike the previous exams, vague ideas might not get marks.
2. While describing the behavior of a Turing machine, it is sufficient to give a high level description, that is, a COL106 or COL351 style algorithm.
3. We have assumed or proved the NP-hardness of only the following problems in class and in Homework 5: `CircuitSAT`, `3SAT`, `Strict3SAT`, `IndSet`, `Clique`, `SubgraphIsomorphism`, `VertexCover`, `SetCover`, `HittingSet`, `DominatingSet`, and the crazy COL352 instructor problem. Only these problems are to be assumed to be NP-hard for this exam.

1. **[1 mark each]** State whether each of the following statements is true or false. Substantiate each answer with a short (ideally, at most 3 sentences of) justification. (Answers with incorrect and insufficient justifications get 0 marks.)
   1. The language $\{r \mid r$ is a regular expression over $\{0,1\}$ and $\mathcal{L}(r) = \{0,1\}^*\}$ is decidable.
   2. If L_1 is a context-free language and L_2 is a regular language then $L_1 \setminus L_2$ and $L_2 \setminus L_1$ are both necessarily context-free.
   3. The languages $\{w \mid \mathcal{L}(M_w)$ is empty$\}$, $\{w \mid \mathcal{L}(M_w)$ is finite$\}$, $\{w \mid \mathcal{L}(M_w)$ is decidable$\}$, $\{w \mid \mathcal{L}(M_w)$ is enumerable$\}$ are all undecidable. (Refer to Homework 4 for the definition of an enumerable language).
   4. `4SAT` (defined analogously to `3SAT`) is Karp-reducible to `3SAT`.
2. Consider the following algorithm which takes a grammar $G = (N, \Sigma, R, S)$ as input and either accepts or rejects it. As usual, N is the set of non-terminals, Σ is the alphabet, R is the set of production rules, and $S \in N$ is the initial non-terminal.

Algorithm 1 Alg(N, Σ, R, S)

$T \leftarrow \emptyset$.
while R contains a production rule of the form $A \to \alpha$, where $A \notin T$ and $\alpha \in T^*$ **do**
  $T \leftarrow T \cup \{A\}$.
end while
If $S \in T$ then Accept, else Reject.

   1. **[4 marks]** What interesting property of G is this algorithm testing? Prove your answer.
   2. **[6 marks]** Make minimal changes to the above algorithm so that your modified algorithm accepts a grammar G if and only if $\mathcal{L}(G) \neq \emptyset$. Argue why your algorithm is correct.
3. **[6 marks]** Consider the following language (which might remind you of `InfVisit` from Quiz 4).

$$\texttt{Visit} = \{(w, x, k) \mid M_w \text{ when run on } x \text{ visits the } k^{\text{th}} \text{ cell of its tape at least once}\}.$$

Is `Visit` recognizable? Is it co-recognizable? Prove your answers. (As usual, M_w denotes the Turing machine whose description is w, and if w doesn't describe any Turing machine legally, then M_w is the Turing machine with two states: $q_{\text{init}} = q_{\text{reject}}$ and q_{accept}, and whose tape alphabet only consists of the input alphabet and the blank.)