CS 581: Theory of Computation
James Hook
Final exam.

This is a closed-notes, closed-book exam.

1. True or False.

   (a) All Regular languages are Context Free.

   (b) The Context Free languages are closed under complement.

   (c) The Turing decidable languages are closed under complement.

   (d) The Turing recognizable languages are closed under complement.

   (e) In a reasonable proof system all true things are provable.

   (f) In a reasonable proof system all provable things are true.

   (g) In a reasonable proof system the set of all provable statements is Turing recognizable.

   (h) In a reasonable proof system the set of all provable statements is Turing decidable.

   (i) Gödel's incompleteness theorem states that for a reasonable theory of arithmetic, including addition and multiplication, there is a provable statement that is not true.

   (j) Gödel's incompleteness theorem states that for a reasonable theory of arithmetic, including addition and multiplication, there is a true statement that is not provable.

2. Prove that ALL_{CFG} is undecidable.

   $$ALL_{CFG} = \{\langle G\rangle | G \text{ is a CFG and } L(G) = \Sigma^\star\}$$

   Hint: This is proved in the text. It is in the section on Reductions via Computation Histories. You may use facts proved in homework problems or supporting facts proved in the text, but you may not assume the result you are asked to show.

   The construction is similar to that of E_{LBA}. You derive a language that is related to the language of computation histories of a machine M on input w. Recall the trick from the $\{x\#y | x \neq y\}$ problem where a CFG can generate two different strings, even when it is impossible to generate two things that are identical.

3. A *virus* is a malicious program that infects a host computer with some malicious logic and has a mechanism for replicating itself and spreading to other hosts. A virus is called *polymorphic* if it changes form as it reproduces.

   This question explores the application of results in this class to the arms race between those who generate malicious code and those who try to detect and subvert it.

(a) What results that we have studied in class suggest that the number of forms a polymorphic virus may take is unbounded? (You may assume that the malicious logic is expressed in a Turing-complete formalism.)

(b) What results that we have studied in class suggest that static checks for polymorphic viruses will necessarily be imprecise, that is they will either have false positives or false negatives?