CS 581: Theory of Computation
James Hook
Final exam.

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

1. (30 points)

   (a) Define a language, A, that is Turing-decidable but not context free.

   (b) Show that A is Turing-decidable. (You may invoke the Church-Turing hypothesis for this and use a programming system other than Turing machines if you wish.)

   (c) Show that A is not context free.

2. (40 points)

   One of the central techniques used in this class is the technique of reduction. We have defined two forms of reduction: mapping reducibility and polynomial time reducibility. We have applied reduction arguments to obtain several key results.

   In this problem you are asked to *sketch* several arguments. Please include clear statements of the reductions required to make your arguments, such as $3SAT \leq_P HAMPATH$. You do not need to give the constructions that justify the existence of these reductions. When possible please use reductions found in the text, lecture, or homework in your sketches.

   (a) Define mapping reducibility.

   (b) Define polynomial time reducibility.

   (c) State and briefly justify the theorem used to relate decidable problems via mapping reducibility.

   (d) State the corollary to the above theorem that is used to prove problems undecidable. Illustrate this reasoning principle with a high level sketch of an example from the text, lecture, or homework. (See note above for interpretation of sketch)

   (e) State and briefly justify the theorem used to relate Turing-recognizable problems via mapping reducibility.

   (f) State the corollary to the above theorem that is used to prove problems not-Turing-recognizable. Illustrate this with an example, such as one of the reductions necessary to show EQ_{TM} is neither Turing-recognizable nor co-Turing-recognizable in the text and lecture.

   (g) State and briefly justify the theorem used to relate polynomial time solutions to languages by polynomial time reduction.

   (h) State the theorem used to relate NP complete problems by polynomial time reductions. Illustrate with an example.

3. Animate Cook-Levin. (30 points)

   This problem illustrates aspects of the proof of the Cook-Levin theorem with an example. In particular, it focuses on the constraints on the 2×3 windows used to verify the integrity of the tableau.

   Consider the nondeterministic Turing Machine (This sample machine is similar to one from a previous exam; the last transition has been changed from a right move to a left move. The question, however, is new.):

$$M = (Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r)$$

   Where

$$\begin{array}{lcl} Q & = & \{q_0, q_1, q_a, q_r\} \\ \Sigma & = & \{0, 1\} \\ \Gamma & = & \Sigma \cup \{b\} \\ \delta(q_0, 0) & = & \{(q_0, 0, R), (q_1, 0, R)\} \\ \delta(q_0, 1) & = & \{(q_0, 1, R)\} \\ \delta(q_1, 0) & = & \{(q_r, 0, R)\} \\ \delta(q_1, 1) & = & \{(q_a, 1, L)\} \end{array}$$

   (a) Construct the 4 by 6 tableau witnessing acceptance of 001.

   (b) Identify the good windows that relate the first row of the table to the second, and the third to the fourth.

   (c) Give an example of a window that would be bad for this machine.

   (d) Propose a set of rules that classify the good windows you enumerated above. These rules do not need to be complete for all machines, but they should classify all windows in your example.

   (e) Argue that your bad window is not allowed by any of your rules.

   (f) Discuss why the proof uses 2×3 windows. Construct an example with 2×2 windows that, in order to accommodate all correct computations, must also allow an incorrect computation. You may want to base your example on M.