18.404/6.5400 Michael Sipser
Theory of Computation
Due via Gradescope at noon sharp, **Tuesday**, November 28, 2023 **Problem Set 5**

Read all of Chapter 8.

1. Recall the language $D = \{\langle p\rangle \mid p(x_1,\ldots,x_m)$ is a polynomial and $p(x_1,\ldots,x_m)=0$ is solvable when all x_i are integers$\}$. We stated, but didn't prove, that D is undecidable. In this problem, you are to prove that D is NP-hard. A language is ***NP-hard*** if all languages in NP are polynomial time reducible to it, even though it may not be in NP itself. You may assume that $\langle p\rangle$ is an expression using plus, minus, multiplication, and constant powers of variables.

2. ***Integer Programming*** is a classic optimization problem which seeks the solution to a system of l linear inequalities on the m integer-valued variables $x_1,\ldots,x_m$.

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1m}x_m & \geq & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2m}x_m & \geq & b_2 \\ & & & & \vdots & & & & \\ a_{l1}x_1 & + & a_{l2}x_2 & + & \cdots & + & a_{lm}x_m & \geq & b_l \end{array}$$

   We can write this system equivalently in matrix form $AX \geq B$ where A is a given $l \times m$ matrix of integers (a_{ij}), B is a given column of l integers (b_i), and X is a column of m unknown integers (x_j). By using the standard matrix product we obtain the inequalities $\sum_j a_{ij}x_j \geq b_i$, for each i. (It's helpful to remember that $c \leq d$ iff $(-c) \geq (-d)$.)

   Here we consider the decision problem of testing whether a given integer programming problem instance has a solution. This problem is NP-complete. Proving its membership in NP is surprisingly difficult, so here we ask you to prove only that this problem is NP-hard. More precisely, let $\mathit{INTEGER\text{-}PROG} = \{\langle A, B\rangle \mid$ some X exists so that $AX \geq B\}$. Here A is an $l \times m$ matrix of integers and B and X are columns of l and m integers respectively.

   Show that $\mathit{INTEGER\text{-}PROG}$ is NP-hard. You may describe your reduction either in matrix form by giving a matrix A and column B, or by giving the equivalent system of inequalities.

3. Let A be a finite set and let $F = \{f_1,\ldots,f_k\}$ be a finite collection of functions $f_i\colon A \longrightarrow A$. Let $C(F) = \{f \mid f = g_1 \circ g_2 \circ \cdots \circ g_l$ where each $g_i \in F$ and $l > 0\}$. (Here $\circ$ denotes function composition so $(g \circ h)(a) = g(h(a))$.) Let $B = \{\langle A, F, t\rangle \mid t \in C(F)\}$. Show that $B \in$ PSPACE.

4. Let B be the language of properly nested parentheses and brackets. For example, `([()()]()[])` is in B but `([)]` is not. Show that B is in L. (Hint: Solve it for just parentheses first.)

5. Show that $E_{\mathsf{DFA}} = \{\langle A\rangle \mid A$ is a DFA and $L(A) = \emptyset\}$ is NL-complete.

6. We defined log-space transducers to have a one-way write-only output tape on which the head can move only left to right. Suppose we modified the definition to allow a two-way write-only output tape on which the head is bidirectional and can write over previously written symbols. Show that the modification doesn't increase the power of the model. In other words, prove that both models define the same class of log-space computable functions.

7.* (optional) Let $A \subseteq \mathtt{1}^*$ be any unary language. Show that if A is NP-complete, then P = NP. (Hint: Consider a polynomial time reduction f from SAT to A. For a formula ϕ, let ϕ_{0100} be the formula where variables x_1, x_2, x_3, and x_4 in ϕ are set to values 0, 1, 0, and 0, respectively and simplified. What happens if you apply f to all exponentially many such formulas?)