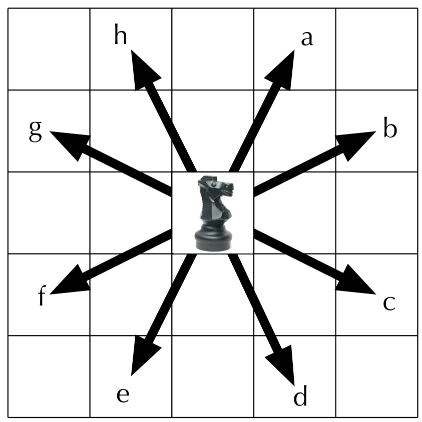
CS500, Theory of Computation: Midterm

Note: Unlike the homework, you may *not* discuss this with other students, nor may you use resources on the Web. You may email me clarifying questions. Due by email in .pdf format by midnight on Sunday, April 18th.

1. (30 points) In chess, a knight can move in eight directions. Label these with eight symbols $\Sigma = \{a, b, c, d, e, f, g, h\}$ as in this figure:



Let $L \subset \Sigma^*$ be the set of paths that return a knight to its original location on an infinite board. Then:

(a) Prove that L is not regular.

(b) Prove that L cannot be recognized in real time by a one-counter machine, i.e.,

i. M reads its input from left to right with no ϵ-transitions.

ii. M has a finite set of states Q. At each step it may update its state and may increment or decrement its counter by any integer.

iii. M's only dependence on the current value of its counter is whether or not it is zero; thus its transition function takes the form $\delta : Q \times \Sigma \times \{\text{zero}, \text{nonzero}\} \to Q \times \mathbb{Z}$ where $\mathbb{Z}$ denotes the integers.

iv. M accepts the input if the counter is zero and/or its final state is in some accepting subset $Q_{\text{accept}} \subseteq Q$.

(c) Prove that L is not context-free.

Note: don't just prove part (c) and then point out that it implies the first two parts! Use proof methods that are "native" to each class of languages or machines, such as closure properties, pumping lemmas, inequivalent states, etc.

2. (20 points) Let L be the set of descriptions of Turing machines M such that M accepts an infinite number of different input strings. Show that L is undecidable. Is L recursively enumerable? Explain.

3. (30 points) In the little-known 51st state, each district contains 3 voters, and the districts may overlap. Political parties there try to win all the district elections by convincing a majority of voters in each district to vote with them. Of course, your party has a limited budget, and you can only afford enough ads to convince up to k voters out of the entire population n, so this might be tricky. Formally:

   District Elections
   Input: An integer n, an integer k, and a set of m subsets $S_1, \ldots, S_m \subset \{1, \ldots, n\}$ such that $|S_i| = 3$ for all i.
   Question: Does there exist a subset $V \subseteq \{1, \ldots, n\}$ with $|V| \leq k$ such that $|S_i \cap V| \geq 2$ for all i?

   (a) Prove that District Elections is NP-complete.

   (b) Exact District Elections is a variant of this problem which asserts that I can win all the elections if I convince the right set of k voters, but that I can't do it with fewer than k (as opposed to District Elections as defined above, which asserts that I can win with k or fewer). Explain why Exact District Elections might not be in NP.

   (c) However, show that if I am given an oracle which answers yes or no to District Elections as defined above, then I can solve Exact District Elections with a reasonable number of calls to this oracle. How many calls do I need as a function of the input size?

   Note: you can do parts (b) and (c) even if you have trouble with (a).

4. (20 points) *Geography* is a two-player game played on a directed graph where we take turns deciding which edge to follow. Whoever gets stuck in a "dead end", with no outgoing edge to follow, loses. The problem Geography is whether, given a directed graph G and an initial vertex u, the first player has a winning strategy. With the restriction that we cannot visit a vertex twice, Geography is PSPACE-complete.

   Now consider removing this restriction. Show that the resulting version of Geography is in P. Hint: find a way to iteratively label vertices of G as "winning" or "losing" vertices, i.e., places you want to be, or places you want to avoid. (Since we can visit the same vertex many times, it might be the case that neither player has a winning strategy and optimal play leads to a draw, i.e., an infinite or cyclic game.)

   Finally, try to explain why the can't-visit-the-same-vertex-twice restriction makes the difference in complexity between P and PSPACE.