Limits of Computation (CS:5340:0001)
Final – Question Types

The exam will be closed book. We will focus on the stuff that was not the subject of the midterm. That is, Section 2.5 and later, all the way upto Boolean Circuits (Chapter 6). We will not have questions on Randomized Algorithms and Complexity Classes (Chapter 7). Here are some possible types of questions we can expect to see and things to review.

1. Definitions of coNP, EXP, and NEXP.
2. Why is coNP contained in EXP?
3. Proof of Theorem 2.22: P = NP implies EXP = NEXP.
4. Decision vs. Search – Problem 1 of Homework 4.
5. Other problems in Homework 4 and their variants.
6. Proof of the time hierarchy theorem.
7. Idea of Configurations in a time- or space-bounded TM, and how they can be thought of us assignments to a set of boolean variables as discussed repeatedly in class.
8. Proof that if $S(n)$ is $\Omega(\log n)$, then any language contained in SPACE($S(n)$) or in NSPACE($S(n)$) is contained in DTIME($2^{O(S(n))}$).
9. Small space algorithms for some languages/problems – determining if a string has an even number of 1's, determining if the length of a given string is a power of 2, etc.
10. Algorithm to show that PATH is in NSPACE($\log n$).
11. Problem 1 of Homework 5 – PSPACE-completeness of SPACE TMSAT.
12. Quantified Boolean Formulae. Problem 2 of Homework 5.
13. PSPACE algorithms – algorithm for TQBF.
14. Why are SAT and its complement special cases of (poly-time reducible to) TQBF?
15. Is NSPACE(n) contained in SPACE(n^3)? Is NSPACE(n) contained in DTIME(2^{n^2})?
16. Describe a poly-size circuit family for a specified language. For example, the language with those strings with an odd number of 1's. Problem 4 of Homework 5.
17. Why is it that any language L is decided by a circuit family of size $O(n2^n)$?
18. Explain why UHALT has a poly-size circuit family.
19. What do we know about containments between the various complexity classes we studied?

I will not ask you for the proofs of the bigger theorems – PSPACE-hardness of TQBF, Savitch's Theorem, and the proof that every language in P has a poly-size circuit family. But you should know the theorem statements, and it is good to have a general idea of how the proofs work.