6.045J/18.400J: Automata, Computability and Complexity Prof. Nancy Lynch

Handout 2: Quiz 1 Information

February 16, 2005 *Vinod Vaikuntanathan*

1 General Information on Quiz 1

Quiz 1 will take place during class (11:00am-12:30pm) on Wednesday, February 23. The quiz will cover Chapters 0 and 1 of Sipser's text, "An Introduction to the Theory of Computation", and all material covered in lectures 1-6. You may bring your copy of Sipser's test with you to class to use during the quiz. However, be advised that bringing a copy of the text is not a substitute for studying the text beforehand: you will not have enough time to read portions of the text that you are not already familiar with. You may also bring a two-sided sheet of handwritten notes.

2 Material You Should Know for the Quiz

The following is a list of important topics and ideas we have covered so far in this class.

- **Mathematical Preliminaries**. You should be familiar with the mathematical concepts from Chapter 0. Specifically, you should understand sets, graphs, functions, relations, and mathematical logic.
- **Proof Techniques**. You should be comfortable with basic proof techniques such as proof by induction and proof by contradiction. You should be able to prove simple theorems related to the material in Chapter 1.
- **Deterministic Finite Automata**. You should understand the formal definition of a DFA, as well as the representation of a DFA as a state diagram. You should understand how a DFA computes and what it means for a DFA to accept an input string and to recognize a language.
- **Nondeterministic Finite Automata**. You should understand the formal definition of a NFA, as well as the representation of an NFA as a state diagram. You should understand how an NFA computes and what it means for a NFA to accept an input string and to recognize a language.
- **Regular Expressions**. You should understand the inductive (recursive) definition of a regular expression. You should be able to tell when a regular expression matches a given input string.
- **Regular Languages**. You should understand what it means for a language to be regular. You should be able to produce examples of regular languages. Given an informal description of a regular language, you should be able to create a DFA, NFA, or regular expression that describes the given language.
- **Operations on Regular Languages.** You should know that the class of regular languages is closed under the standard set-theoretic operations and under concatenation and star. You should understand how to prove theorems similar to: "if L_1 and L_2 are regular, then L_1 union L_2 is regular".
- **Equivalence of DFAs and NFAs**. You should understand that DFAs and NFAs recognize the same class of languages. You should be able to convert an NFA into a DFA that recognizes the same language.
- **Equivalence of NFAs and Regular Expressions**. You should understand that NFAs and regular expressions recognize the same class of languages. You should be familiar with Generalized NFAs. You should be able to convert a regular expression into an equivalent NFA and vice versa.
- **Non-Regular Languages**. You should understand what it means for a language not to be regular. You should know several examples of non-regular languages. You should be familiar with the Pigeonhole Principle and the Pumping Lemma. You should be able to prove that simple non-regular languages are in fact non-regular, using either the Pigeonhole Principle or the Pumping Lemma.
- **Algorithms**. You should be able to describe algorithms that answer basic questions about DFAs, NFAs, and regular expressions, for example, whether they accept any words at all, or whether two representations describe the same language.