6.045J/18.400J: Automata, Computability and Complexity Nancy Lynch and Nati Srebro

Handout 7: Quiz 2 Information

March 17, 2004 *Susan Hohenberger*

1 General Information on Quiz 2

Quiz 2 will take place during class (11:00am-12:30pm) on Wednesday, March 31. The quiz will cover Chapters 3, 4, and 5 of Sipser's text (except the context-free material on pages 156-158 and 181-182), as well as Section 6.1 and the supplementary material on Rice's Theorem (handout 6) and counter machines (handout 10). Handouts 6 and 10 are not online; if you do not have a copy, contact the TA. You are responsible for material covered in lectures 8-14 (March 1 to March 29). You may bring your copy of Sipser's text with you to class to use during the quiz, as well as the handouts on Rice's Theorem and counter machines. 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 covered during the computability portion of this course. (Please note that this is a *two*-sided handout.)

- **Turing Machines and the Church-Turing Thesis**. You should understand the Church-Turing thesis. You should know the formal definition of a Turing Machine and understand the way that a Turing Machine computes on a given input. You should know what it means for a Turing machine to accept an input, and to recognize or decide a language. You should be comfortable with using one model of computation to simulate another model of computation (multi-tape Turing Machines, nondeterministic Turing Machines, etc.)

- **Recognizability and Decidability**. You should be able to define both Turing-recognizability and Turing-decidability, and should understand the difference between the two concepts. You should be able to give examples of both recognizable and decidable languages.

- **Enumeration**. You should know what an enumerator is, and should know what it means for a language to be enumerable. You should understand why being recognizable is equivalent to being enumerable.

- **Undecidability**. You should be comfortable with the idea of a Turing Machine computing on inputs that are themselves representations of Turing Machines. You should know what it means for a language to be undecidable and you should understand why the Halting Problem is undecidable. You should know several other examples of languages that are undecidable. You should be comfortable with diagonalization and reduction techniques for proving that languages are undecidable.

- **Mapping Reducibility**. You should understand the formal definition of mapping reducibility as well as the intuition behind this definition. You should be able to use mapping reducibility as a tool for showing languages are either decidable or undecidable.

- **Post Correspondence Problem**. You should be comfortable with the definition of the Post Correspondence Problem. You should know that the PCP is undecidable. You should be familiar with computation histories. You should understand the general techniques used for showing that the PCP is undecidable but you will not be expected to be able to reproduce this proof on the quiz.

- **Stack Machines and Counter Machines**. You should be familiar with stack machines and counter machines. You should understand how to use such machines to simulate general Turing machines. You should know that the acceptance problems for certain kinds of stack machines and counter machines are undecidable, and should understand the general techniques used for showing these problems are undecidable.

- **Rice's Theorem**. You should know the statement of Rice's Theorem and understand when it applies and when it does not apply. You should be able to use Rice's Theorem to show problems are undecidable. You should understand the technique used to prove Rice's Theorem.
- **Recursion Theorem**. You should know the statement of the Recursion Theorem and understand how it is used. You should be familiar with the applications of the Recursion Theorem that we discussed in class.