

Name: _____

Exercise B5 (Module B Comprehensive Exercise), 2/25/05

Note: This exercise is announced in advance so that we can think about it through this module. The due date of this exercise is the date of Module B Evaluation Workshop.

Part 1: Analyzing Computability

One of the main goals of Modules A & B (and the Theory of Computability as a whole) is to be able to identify the computability class of real-world problems (so that we won't write a general program for an undecidable problem). In this module, we have been analyzing the main problems such as the halting problem with respect to their universality, self-reference, countability, reduction, etc. The real test for us is whether we can develop and describe our thought process involved in such activities and apply it to a new problem. While this is a major problem in the Theory of Computation, it is unlikely that you will easily find an answer in the literature (what are computer scientists doing?). By all means, it is a challenge. So, you are not expected to come up with a rock solid answer, not to mention a correct answer. Do your best. As usual, you are encouraged to discuss with other students and the instructor (but your writing must be your own, reflecting your own thought, which must be unique to yourself). If we as a class come up with useful analytical tools, it would be great.

Note: Since **Tasks 1 & 2** are closely related, you should tackle them in parallel, or in a circular manner (back and forth).

Task 1: Develop your own "heuristics" to analyze a given problem with respect to the computability classes, i.e., decidable, semi-decidable, or non-TM-recognizable. By "heuristics," we mean a speculative, but reasonable method of analyzing *any* given problem. For example, one might consider a rule such as: (i) if self-reference is involved, the problem is undecidable, or (ii) if the problem is uncountable, it is undecidable (these may or may not be true or even relevant; they are just examples of how to phrase a component of your method).

Task 2: This task is an application of the previous one. (A) Identify some unique practical problem (in or outside computer science) which would be a good example of demonstrating your heuristics in **Task 1**. (B) Then, provide the set representation of the problem. (C) Finally, apply your heuristics developed in **Task 1** to the problem to analyze its computability class.

Part 2: Evaluation Form and Supporting Notes

Review the evaluation procedure. Then, complete your evaluation form and supporting notes. Bring them to the evaluation workshop (hard copy). Print them well in advance so that you can avoid potential problems, e.g., not being able to print just before the evaluation.

Survey: Time spent between after Unit B5 before the evaluation workshop: _____

// End