

## Module B Review Exercise, 3/1/05

(1) Do the required problems *and* (2) analyze and explain the **computability class** (e.g., decidability) of one of the optional problems, applying your own heuristics. In case your own problem in the comprehensive exercise is similar to any of the optional problems, you must try a different problem. Write your response on back or on another sheet of paper. Note: You are **not** supposed to solve any of the problems.

### Required Problems: '\$' Detection

Analyze and compare the computability classes of the following two problems:

- A. To find out whether a given string contains at least one instance of the symbol '\$'
- B. To find out whether a given TM decides on Problem A (above)

### Option 1: Virtual Memory Paging (Architecture)

To come up with an algorithm to find out whether page fault is avoidable, given a program, memory specification, and the specification of the applicable machine architecture

### Option 2: Deadlock Prevention (OS)

To come up with an algorithm to prevent deadlocks, given a set of processes (which may spawn subprocesses) and resources (which may become unavailable)

### Option 3: Optimization (Compilers)

To design a procedure to generate the most optimized target code of the given source code with respect to the target code size

### Option 4: Protocol Security (Networks)

To develop a process to find out whether a given network protocol is secure

### Option 5: Semantic Errors (Databases)

To identify a mechanism to detect semantic errors (e.g., contradictory queries, which would return an empty set) in database queries

### Option 6: Planning (AI)

To develop a program to construct a plan (of actions) that would lead to the desired goal state, given a set of primitive actions

### Option 7: Best Solution

To find the best solution, given a problem and some specification about the goodness of a solution

### Option 8: Decidability Classes

To identify the computability class of a given problem

// End