Name: ______________________

Exercise C5 (Module C Comprehensive Exercise), 3/29/05

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 C Evaluation Workshop.

Part 1: Identifying the Simplest Mechanism Along the Extended Chomsky Hierarchy

One of the main goals of Modules C is to be able to identify the minimal mechanisms for real-world problems (so that we won't use insufficient or overly expensive models). In this module, we analyze several representative problems, e.g., lexer, parser, and variable co-reference in a compiler. 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 another major problem in the Theory of Computation, it is unlikely that you will easily find an answer in the literature. 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 minimal mechanism, i.e., decidable, context-free, deterministic context-free, regular grammar/automaton/language. By "heuristics," we mean a speculative, but reasonable method of analyzing *any* given problem.

Task 2: This task is an application of the previous one.

(A) First, identify a new, unique practical problem (in or outside computer science) which would be a good example of demonstrating your heuristics in **Task 1**. Then, identify multiple interactive subproblems/modules of your problem. If the problem is complex, you can focus on the areas which you are most interested in. But still try to include modules of different complexity. Also, try to represent the system schematically.

(B) Next, for each subproblem/module,

1. Analyze its task and clearly present it as a problem (if possible, as a computational problem in the set notation),
2. Applying your heuristics developed in **Task 1**, identify the class of the minimal grammar or automaton that would correspond to the problem (mainly consider the extended Chomsky hierarchy: i.e., decidable, context-free, deterministic context-free, or regular), *and* justify your choice, and
3. Describe the component's specification or operation at a reasonable level.

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 C5 before the evaluation workshop: _______________

// End