CSC460 (Spring 2005) Module C Evaluation Form

Name	Self-evaluation (A, B, C possibly with +/–)	
	Adjustment by the instructor	

Evaluation Materials (Portfolio)

Your evaluation materials (referred to as "**portfolio**," and to be placed in the provided manila folder) consist of the following **Items**:

1. This form (must be filled out; see the instructions below)
2. Word-processed **supporting notes** responding to the instructions in this form (except for the materials completed during the evaluation workshop)
3. Take-home **exercises** (including the comprehensive exercise), chronologically ordered
4. Materials completed during the **evaluation workshop** (to be explained in class)

Learning Goals Checklist (the goals not pursued in this module are "grayed")

In your **supporting notes**, clearly identify the criteria, e.g., **C1*a*** (for Content Goal 1 Criterion *a*), **P5*b*** (for Performance Goal 5 Criterion *b*), referring to the labels below.

Content Goals

1. Practical problems can often be transformed into research and computational problems. Every problem is associated with cost/significance, which is relative to the evaluator. Computational problems can be represented as a set, readily available as the input to computational processing. [**problem**] [not in this module]
2. A theory is a potentially infinite, consistent body of knowledge which can be systematically derived from a small number of abstract principles. The gap between the principles and the entire information is the source of the theory's predictive power. Also being abstract, a theory can be applied to a broad range of phenomena, which might appear distinct. [**theory**] [not in this module]
3. Interactive computation subsumes algorithmic computation, but not vice versa. That is, there is a qualitative difference between these two modes of computation. [**interactive computation**] [not in this module]
4. The algorithmic notion of computation can be represented in a variety of equivalent forms, which define a bounded class of sets. That is, there is a limit to algorithmic computation. [**computability**] [not in this module]
5. The computable class of sets contain a hierarchy of proper subsets which can be characterized by distinct grammars and automata. [**formal languages and automata**]
   - *a.* Understood that there is a hierarchy of TM-recognizable languages that can be specified and processed by the corresponding grammars and automata, respectively [Chomsky hierarchy]. ☐
   - *b.* Understood that TM-recognizable languages can be specified by unrestricted grammars (rewriting system), which are equivalent to TMs. Also understood why this class is not useful in practice. ☐
   - *c.* Understood that context-free languages (CFLs) can be specified by context-free grammars (CFGs) and processed by push-down automata (PDAs). Also understood when/how to use this class to analyze problems. ☐
   - *d.* Understood that the deterministic subset of CFLs is an important class that supports the backbone of programming languages, which can be processed by deterministic PDAs (DPDAs). ☐
   - *e.* Understood that regular languages (regular sets) can be specified by regular expressions (RegExps) and processed by finite-state automata (FSAs). Also understood when/how to use this class to analyze problems. ☐
   - *f.* Understood how to use the pumping lemma to show that a language is *not* regular. ☐
   - *g.* Understood how to use the pumping lemma for CFLs to show that a language is *not* context-free. Also understood that certain properties of CFLs are undecidable. ☐
   - *h.* Could explain (i.e., teach) this goal to CS students outside this class. ☐
6. The practicality of an algorithm depends on its complexity relative to the input data size. [**complexity**] [not in this module]
7. Power set, also encompassing the distinction between determinism and nondeterminism, can introduce discontinuity with respect to computability and complexity. [**power set**]
   - *a.* Understood that nondeterminism can be represented as the power set of the possible states. ☐
   - *b.* Understood why nondeterminism affects the power of PDAs but neither TMs nor FSAs. ☐

Performance Goals

1. Identify real-world problems which are relevant to the student's life and can be tackled by computational means. [**awareness**] .... [combined with other goals]
2. Transform real-world problems into research problems, and then into computational problems, along with the analysis of the cost/significance of a problem. [**transformation**] .... [not in this module]
3. Analyze computational problems with respect to interactivity, computability, language/automata hierarchy, and complexity hierarchy. Then, evaluate the analysis with respect to its usefulness, correctness, and accuracy. [**analysis/evaluation**] .... [combined with content goals]
4. Respect, analyze, and give constructive criticisms to the ideas in the literature and those expressed by other people. [**critical attitude**]
   a. *Critically* analyzed the usefulness of the "languages/automata" area of the (traditional) Theory of Computation, esp. in connection to your ability to choose the minimal specification/process for the given problem. .... ☐
5. Express ideas orally and in writing, in a manner clearly understood by other students (with equivalent background). Explain your own ideas orally and in writing, clearly and logically. Revise the ideas, reflecting the feedback from other students and the instructor. [**communication**]
   a. Completed all the exercises on time (take-home and in-class). .... ☐
   b. Made conscious efforts to promote transfer of learning among students, esp. during in-class exercises (e.g., explained what you understood to other students and learned what you didn't understand from other students). .... ☐
6. Take initiative in both independent and group activities. Also extend the domain of theoretical inquiry beyond the scope prepared by the instructor. [**initiative**]
   a. Continued to analyze your own problem and mini research project, by applying the notion of "modularity" and identifying the simplest mechanisms for the component modules. .... ☐
   b. Noted any other aspects relevant to this goal. .... ☐
7. Reflect upon the student's own thinking process and assess the student's own performance relative to the content and performance goals. [**reflection**]
   a. Was able to reflect upon your experience in this module through the activities during the evaluation workshop. .... [during the eval workshop]
   b. Self-evaluated your achievements *accurately*. .... [during the eval workshop]

Self-Evaluation Criteria

At the end of the module evaluation workshop, propose your grade based on the following scheme (possible qualification with +/–):

- Grade A: Achieved all the learning goals relevant to the module
- Grade B: Achieved almost all the learning goals (except for one or two evaluation criteria) relevant to the module
- Grade C/Pass: Achieved most of the learning goals relevant to the module

// End