CSE 425S Final Exam

DUE: 9:00 am Monday, 15 December 2008 in the CSE office (Bryan 509)

Certification:

“I certify that I did not discuss the contents of this exam with anyone, except the professor and the teaching assistant for this class. I understand that any form of collaboration on this exam represents an integrity violation serious enough so as to warrant my failing the class.”

(signature)

(full name)

Special instructions:

a. Place these three pages at the beginning of your take home exam.

b. Do not repeat the questions in the answer section of the exam.

c. Use 10pt font size and single spaced paragraphs.

d. Limit answers to at most one page each, excluding any figures.
0. Course evaluation (10 pt)

   a. Did you complete the online course evaluation? There is only one right answer!

1. Syntax (15 pts)

   a. Give the BNF specification for a restricted class of Lisp-like cond statements in which all
      conditions involve integer comparisons and the returned values are integers as in the example:
      
      (cond ((< x 0) -1) ((= x 0) 0) ( (> x 0) -1))

   b. Explain the concept of a recursive-descent parser.

   c. Sketch out a recursive descent parser using the syntax you specified.

2. Axiomatic semantics (15 pts)

   a. Prove the correctness of the following statement with respect to the assertions provided below
      
      \{ x < y \}
      
      if x < y then t := x; x := y; y := t fi
      
      \{ x > y \}

3. Dangling references and garbage (15 pts)

   a. Consider a generic block-structured language with pointers. What restrictions would you
      impose on the language definition in order to avoid dangling references and the generation of
      garbage?

4. Type constructors (15 pts)

   a. New programmer-defined types are usually built using type constructors. Many of the
      constructors in use today are based on the set concept. Propose a suite of type constructors
      based on the mathematical notion of function. Suggest a possible syntax for each constructor.

5. Parameter passing (15 pts)

   a. Define the terms call-by-reference and call-by-name.

   b. Write a very simple program, which allows you to illustrate the difference between call-by-
      reference and call-by-name.

6. Activation record (15 pts)

   a. Consider a generic block-structured language. Upon entering a procedure, an activation record
      is created and is placed on the top of the stack. What information is included in the activation
      record?

7. Abstract data type specification (20 pts)

   a. Give a formal specification for the set abstract data type.

8. Storage management (20 pts)

   a. Consider a run time system in which all storage allocation is in terms of blocks whose size is a
      power of two, e.g., 2k, 4k, 8k, etc. Assume explicit storage space release. Propose an efficient
      way to manage the storage reclamation process.
9. **Synchronous communication (30 pts)**

a. Consider a language in which a program consists of a fixed finite set of named concurrent processes and communication takes by matching send and receive statements from corresponding processes. For instance, the code fragment

\[
\text{process } \text{P} \begin{align*}
\text{begin} & \quad \ldots (Q?x \text{ or } R!y) \ldots \text{end}
\end{align*}
\]

indicates that process P is willing to receive a value to be stored in x from process Q or to send the value stored in y to process R, but not both. If neither communication is feasible at that time, process P will wait until at least one of the two communication statements can be executed. When multiple send/receive statements are ready to execute, the runtime system will select one non-deterministically.

b. Sketch out the key elements of a runtime system that implements this communication policy, i.e., the data structures it needs to maintain in order to manage the communication process.

c. Assume that the runtime system makes scheduling decisions only when a communication statement is executed, i.e., a running process that executes no communication statements will not allow any other process to run until such time that itself terminates. Propose a simple language construct that might allow such a process to release control to the runtime system explicitly even though it does not need to communicate with any other process at that time. Your choice should have little or no impact on the design of the runtime system.

10. **Concurrency management (30 pts)**

a. Consider a very simple sequential block structured language that supports procedures and integer types. Extend the language by introducing the construct

\[
\text{run } \text{P}
\]

where P is a defined procedure visible at that point in the program.

b. What new runtime structure needs to be added to manage the execution of multiple concurrent processes?

c. What is the impact, if any, on the definition of the procedure activation record?

d. What is the impact, if any, on the run time stack?