Software Design CS2ME3 Michael Soltys

DAY CLASS
DURATION OF EXAMINATION: 3 Hours
MCMASTER UNIVERSITY FINAL EXAMINATION April 28, 2004

THIS EXAMINATION PAPER INCLUDES 3 PAGES AND 10 QUESTIONS. YOU ARE RESPONSIBLE FOR ENSURING THAT YOUR COPY OF THE PAPER IS COMPLETE. BRING ANY DISCREPANCY TO THE ATTENTION OF YOUR INVIGILATOR.

No aids allowed.

All questions worth 10 marks, for a total of 100.

1. This question is about the basic concepts of program development.
   (a) List the six desirable program characteristics. Describe each very briefly.
   (b) List the four basic activities of any software development effort. Again, describe each very briefly.
   (c) Explain the Software Life Cycle.
   (d) Describe the five models of software development, in order of increasing sophistication. Give the advantages and disadvantages of each of them.
   (e) Choose two basic software testing concepts, and describe them very briefly.

   You are encouraged to draw diagrams to answer (c) and (d).

2. In our course we were very concerned with program correctness. Explain, giving one example of a spectacular software failure, why software correctness is absolutely crucial in many cases.

3. Define the following concepts.
   (a) Specifications.
   (b) Precondition and Postcondition.
   (c) Loop invariant.
   (d) Hoare's Logic.

4. We showed in class that it is possible to prove the correctness of a piece of **C** code formally, using the rules for program verification plus LK-PA. Even for a short program, this can be a very lengthy and tedious proof, which can be *partly* automated using some verification software. Explain why partly automated, and not fully. What is the advantage of producing such a long tedious proof of correctness; who needs it?

**Continued on Page 2**

5. Give a precondition and a postcondition for the **C** code presented below. Choose your precondition and postcondition so that the intended correctness of the program follows from them.

   Then, give a loop invariant, prove that the loop invariant holds, and show that the postcondition follows from the precondition and the loop invariant.

```
int power(int x) {
    int current = 1;
    int count = 0;
    while(count < x) {
       current = current*2;
       count += 1;
    }
}
```

6. The Goldbach Conjecture states that every even integer greater than 2 is the sum of two primes. States this conjecture as an $\mathcal{L}_A$-sentence. Recall that $\mathcal{L}_A = [0, s, +, *; =]$ is the standard language of arithmetic.

7. This question is about the system LK-PA defined in class.

   (a) State all the rules of LK.

   (b) State the axioms P1-6 of Peano Arithmetic, the Induction Axiom Scheme, and the equality axioms E1-5.

   (c) Consider the following sequent:

   $$\rightarrow \forall x \forall y \forall z (x + (y + z)) = ((x + y) + z)$$

   which states that addition is associative. Give an <u>outline</u> of the LK-PA derivation of this sequent. You do not have to give the entire derivation tree, but rather outline the main parts of the derivation, pointing out which axioms are being used. The clarity of you answer will be a big component of the mark.

   **Hint.** You will need the Induction Axiom. Suggestion: do induction on $z$.

8. A monotone propositional formula is a formula whose only connectives are $\wedge$ (AND) and $\vee$ (OR). (Thus, the set of monotone propositional formulas is the set of propositional formulas without $\neg$ (NEGATION).)

   Suppose that you are asked to write a parser that recognizes exactly the set of monotone propositional formulas. Explain (in some detail) how to construct such a parser.

   **Hint.** Make sure that your answer contains the following "key-words": context-free grammar, Chomsky normal form, CYK algorithm.

**Continued on Page 3**

9. The Make-Change Problem consists in paying a given amount using the least number of coins, using some fixed denomination, and an unlimited supply of coins of each denomination.

   Consider the following greedy program to solve the change making problem, where the denominations are $\{1, 10, 25, 100\}$:

   list MakeChange($N$) * *makes change for $N$ units using coins from $C$* *\
       const $C \longleftarrow \{1, 10, 25, 100\}$
       $L \longleftarrow \emptyset$         * *$L$ is the list holding the solution* *\
       $s \longleftarrow 0$         * *$s$ is the sum of the items in $L$* *\
       while $(s < N)$
           $x =$ largest item in $C$ such that $s + x \leq N$
           $L \longleftarrow L, x$
           $s \longleftarrow s + x$
       return $L$

   (a) Show that this greedy program (with the given denominations) does *not* necessarily produce an optimal solution. That is, present an $N$ for which the output $L$ of MakeChange($N$) contains more coins than the optimal solution.

   (b) Suppose now that the available denominations are $\{1, p, p^2, \ldots, p^n\}$, where $p > 1$ and $n \geq 0$ are integers. That is, "$C \longleftarrow \{1, 10, 25, 100\}$" in the above algorithm is replaced by "$C \longleftarrow \{1, p, p^2, \ldots, p^n\}$." Show that with this series of denominations (for some fixed $p, n$) the greedy algorithm above always finds an optimal solution. (**Hint.** Start with a suitable definition of a promising list.)

10. Consider the following variation of the Longest Monotone Subsequence problem:

    **Input:** $d, a_1, a_2, \ldots, a_d \in \mathbb{N}$.

    **Output:** What is the length of the longest subsequence of $a_1, a_2, \ldots, a_d$, where any two consecutive members of the subsequence differ by at most 1?

    For example, the longest such subsequence of $\{7, 6, 1, 4, 7, 8, 20\}$ is $\{7, 6, 7, 8\}$, so in this case the answer would be 4.

    Give a dynamic programming solution to this problem by doing the following three steps:

    (a) Define an array $R$ of subproblems. Explain how to obtain the solution from $R$.

    (b) Find a recurrence (including initial conditions) that could be used to compute the values in the array $R$.

    (c) Write a (pseudo-code) program for computing the values in the array $R$.

## The End