Software Design I (CS2ME3) Michael Soltys

DAY CLASS
DURATION OF EXAMINATION: 2 Hours
MCMASTER UNIVERSITY FINAL EXAMINATION April 2006

THIS EXAMINATION PAPER INCLUDES 2 PAGES AND 5 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 50.

1. Recall **Kruskal's Algorithm:**

   Sort the edges in nondecreasing order of costs
   $T \longleftarrow \emptyset$
   for $i : 1..m$
       if $T \cup \{e_i\}$ has no cycle then
           $T \longleftarrow T \cup \{e_i\}$
       end if
   end for

   Suppose that the costs of all the edges are distinct; that is, after ordering the edges by costs we have $c(e_1) < c(e_2) < \cdots < c(e_m)$. Show that in that case $G$ has a *unique* minimum cost spanning tree.

2. Recall the **All Pairs Shortest Path Problem**:

   **Input:** Directed graph $G = (V, E)$, $V = \{1, 2, \ldots, n\}$. Cost $C(i, j) \in \mathbb{R}^+ \cup \{\infty\}$ (think of the costs as distances), $1 \leq i, j \leq n$, $C(i, j) = \infty$ if $(i, j)$ is not an edge.

   **Output:** $D(i, j)$, the length of the shortest directed path from $i$ to $j$.

   Give a dynamic programming solution to this problem. Make sure that you specify clearly the three steps in the design of a dynamic programming solution.

**Continued on Page 2**

3. Consider the following program for integer division (which takes as input $x, y$ and outputs the quotient and reminder of the divisition of $y$ by $x$):

$q \leftarrow 0$
$r \leftarrow x$
while $(y \leq r)$
  $r \leftarrow r - y$
  $q \leftarrow q + 1$
end while

Provide an appropriate pre and post-condition. Define a loop invariant and show correctness (that is, show that your loop invariant indeed holds, and that the post-condition follows from the pre-condition and your loop invariant). Finally, show that the program terminates.

4. Define the system PK for deriving valid sequents. Using PK, design an algorithm for checking the validity of sequents, that is, an algorithm $\mathcal{A}$ which takes as input a sequent $S$, and outputs "yes" if $S$ is valid, and "no" otherwise. Furthermore, when $S$ is *not* valid, $\mathcal{A}$ should output a truth assignment that falsifies $S$. (**Hint.** Your algorithm should attempt to construct a derivation of $S$, working backwards from $S$; when do you stop?)

5. Define the system LK-PA (provide the language of PA, $\mathcal{L}_{\text{PA}}$, as well as the rules and axioms of LK-PA). Then show that LK-PA proves that every non-zero element has a predecessor, that is, show that LK-PA proves the sentence $\forall x(x = 0 \vee \exists y(x = sy))$.

## The End