COP5025-1 Spring 1999 – Final Exam (Chs. 8–9)

Name: _____ (Please print)

You can put the answers on these sheets. Use additional sheets when necessary. Always show how you derived your answers (this is required for full credit and helpful for partial credit). You can collect 100 points in total for this exam. A bonus question is included for an additional 15 points.

1. Consider the language of simple expressions (Section 8.6; transparencies: example 4). What functions do the following expressions denote? (15 points)

   (a) $1 + (x - x)$

   (b) **let** $x = y$ **in** $x - y$ **end**

   (c) **let** $y = 1$ **in let** $x = y$ **in** $x + y$ **end end**

2. Consider the language with commands (Sections 8.8 & 8.9; transparencies: examples 6 & 7). Suppose we add an if-then command to the language:

   $C \rightarrow$ **if** E **then** C

   Give a definition for the semantic function for the if-then construct. (15 points)

3. The textbook has some annoying typo's. One of the "interesting" typo's is on page 281:

$$\mathcal{C}[\![\textbf{while } E \textbf{ do } C]\!]\ \sigma = \begin{array}{ll}\texttt{let} & \mathrm{p}(\sigma') = \texttt{if } \mathrm{IsTrue}(\mathcal{E}[\![E]\!]\ \sigma) \texttt{ then } \mathrm{p}(\mathcal{C}[\![C]\!]\ \sigma') \texttt{ else } \sigma' \\ \texttt{in} & \mathrm{p}(\sigma) \\ \texttt{end} & \end{array}$$

Compare this definition to the correct one on the transparencies:

$$\mathcal{C}[\![\textbf{while } E \textbf{ do } C]\!]\ \sigma = \begin{array}{ll}\texttt{let} & \mathrm{p}(\sigma') = \texttt{if } \mathrm{IsTrue}(\mathcal{E}[\![E]\!]\ \sigma') \texttt{ then } \mathrm{p}(\mathcal{C}[\![C]\!]\ \sigma') \texttt{ else } \sigma' \\ \texttt{in} & \mathrm{p}(\sigma) \\ \texttt{end} & \end{array}$$

The semantic function of the loop is a recursive function p that takes some machine state σ' and returns the new state after executing the loop. Consider for example

```
n := 7;
while n do
  n := n-1
end
```

The machine state before entering the loop contains $\sigma(\rho(\mathtt{n})) = 7$. The function application $\mathrm{p}(\sigma)$ should return the new state after the loop.

What do you think will happen when the incorrect semantic function definition in the book is used to define the semantics of the example program containing the loop? (Do not apply the semantic functions to get the semantic denotation, but give an explanation of the behavior of the program). (20 points)

4. Which of the following Hoare triples are valid? (Show derivations) (20 points)

   (a) $\{x = 0\}$ `if` $x = 0$ `then` $y = x$ `else` $x = y$ $\{x = 0\}$

   (b) $\{x = 3\}$ $x := 2 * x + 1$ $\{x = 9\}$

   (c) $\{y = 0\}$ $x := x * y$ $\{x = 0\}$

   (d) $\{x = 1 \,\&\, y = 2\}$ $(x, y) := (y, x + y)$ $\{x = 2 \,\&\, y = 3\}$

5. You are employed in company Softmicro. The company boss (#!+?¡@&¿*) tells you to write a program to break the encoding of the Scapenet browser code. The encryption key is a large integer, so the program may run for a long time. With this information you wrote the following program:

```
key := 0;
while found <> 1 do
  if is_key(key) then found := found+1 else key := key+1
end
```

As you demonstrate this program to your boss, one of Murphy's programming laws comes into effect and the program does not terminate! The boss is angry after several years of waiting and wants to fire you because of the bug. However, you lookup the company's definition of a bug and you find:

A. bug = an insect

B. bug = program that is not partially correct

Suppose that the postcondition of the program is $found = 1$. Answer the following:

(a) By this definition, does the program contain a bug? Explain. (10 points)

(b) What is the precondition of the program to ensure that it terminates given that at least one key exists? (Do not use a formal proof, give an informal explanation) (10 points)

(c) What is the precondition of the program to ensure that it terminates given that there is no key at all? (Do not use a formal proof, give an informal explanation) (10 points)

6. **Bonus question.**(15 points)

Consider the following program segment:

```
f1 := 0;
f2 := 1;
n := x;
while n>0 do
  (f1, f2) := (f2, f1+f2);
  n := n-1
end
```

(Note the use of the simultaneous assignment statement in the loop).

This program computes the Fibonacci number of x, that is, $f1 = \text{fib}(x)$ after execution. Let the loop invariant I be $f1 = \text{fib}(x-n) \,\&\, f2 = \text{fib}(x-n+1) \,\&\, x \geq n \,\&\, n \geq 0$. The postcondition is $I \,\&\, 0 \leq n$ which simplifies to $f1 = \text{fib}(x) \,\&\, f2 = \text{fib}(x+1) \,\&\, x \geq 0 \,\&\, n = 0$. Prove the correctness of this program segment and derive it's weakest precondition.

You can use the following properties of the *fib* function:

$$\begin{aligned} \text{fib}(0) &= 0 \\ \text{fib}(1) &= 1 \\ \text{fib}(x-n+2) &= \text{fib}(x-n) + \text{fib}(x-n+1) \qquad \text{with} \quad x \geq n \,\&\, n \geq 0 \end{aligned}$$