Programming Language Comprehensive, September, 1995

INSTRUCTIONS. Do all 6 questions. You have 2 hours.

Question 1: Here is a program that sets x to the number of zeroes in an array segment $b[m]...b[n]$. Don't worry about bounds of the array; we will assume that it contains the segment. The program contains a pre and postcondition. Your job is to provide a loop invariant. Explain how the loop invariant you give combines with (assumption of) termination of the loop to imply the postcondition.

```
{n>=m}
x:=0;
i := m;
{Invariant:
while i <= n do
begin
  if b[i] = 0 then x:=x+1;
  i:=i+1;
end
{x = number of zeroes in interval b[m]...b[n]}

$\;$\\
$\;$\\
$\;$\\
$\;$\\
$\;$\\
$\;$\\
$\;$\\
Give a (slightly) different algorithm which leads to a simpler
invariant.
```

Question 2: Define, in a few lines, the concepts of *state* and *environment*. What is the essential difference between these concepts? What kinds of programming language constructs affect the state, and what ones affect the environment?

Name a language that confuses (fails to distinguish) the state and environment, and one for which the concepts indeed are separate.

Question 3: (A) State Hoare's axiom for assignment statements, and give two instances of it.

(B) Give the rule for sequential composition

(C) Give a derivation in Hoare's logic of

$$\{x = 4\}\, x := x + 1; y := x * 2\, \{y = 10\}$$

Question 4.

Consider a typed call-by-name functional language, with types

$$t ::= \text{integer} \mid t \to t$$

(A) Here is a recursive definition of a factorial function of type integer $\to$ integer

```
factorial(n) = if n=0 then 1 else n*(factorial(n-1))
```

The semantics of this is given as the least upper bound of a sequence of approximations f_i to factorial. What are these functions f_i?

(B) What is the type of the least fixed-point operator Y_t (also sometimes called $rec[t]$)? Rewrite the factorial function without explicit recursion, using λ and Y. What is the type of this instance of Y?.

(C) Suppose each type is interpreted as a domain (complete partially ordered set with least element) $[\![t]\!]$, where $[\![t \to t]\!]$ is the continuous function space. Give the definition of the lease fixed-point operator in terms of these domains.

Question 5. The storage allocation for procedure calls, particularly environment manipulation for parameters, is markedly different in Lisp or Scheme than in conventional procedural languages such as Pascal. What is the key difference in the mechanisms used. Why is it necessary to use a different mechanism for Lisp?

Question 6. (A) What part of the compilation process are regular expressions often used for? What part are context-free grammars often used for?

.

(B) The regular expression $(a|b)^*$ generates a language whose strings are arbitrary sequences of a's and b's. Give a regular expression for this same languagethat does not contain the bar |. Give a regular expression for a language consisting of all strings of a's, b's and c's in which there are an even number of b's.

(C) Define (briefly) the notion of context-free grammar. Give an example of a context-free grammar that has infinitely many strings in the language it accepts. Give an example of a grammar that is not context free.