Logics of Programs	**Take-Home Final**
CS686 Spring 2001	**Due Friday, May 18**

1. Show that in KAT, $bp = pc$ is equivalent to $bp\overline{c} + \overline{b}pc = 0$.

2. Use KAT to prove that the following two programs are equivalent:

```
while b do p;
while c do {
  q;
  while b do p;
}
```

```
while (b or c) {
  if b then p;
  else if c then q;
}
```

3. Prove that $\mu X.\varphi = \mu X.(\varphi[X/\varphi])$. For example, $\mu X.[a]X = \mu X.[a][a]X$. Use the deductive system, do not reason semantically.

4. (a) Give a nondeterministic Büchi automaton over a binary alphabet $\{a, b\}$ accepting the set of complete binary trees such that exactly one node somewhere in the tree is labeled a.

   (b) Show that no deterministic Muller automaton can accept this set. Why does this not contradict Safra's theorem?

5. In branching-time temporal logic, the path quantifier A means, "For all paths starting at the current state, ... ," and $\Box$ and $\Diamond$ have their usual linear-time interpretations on paths. Thus $\mathsf{A}\Diamond\Box\neg p$ says that p is true only finitely often along any path.

   (a) Give an example showing that even in a binary tree, $s \vDash \mathsf{A}\Diamond\Box\neg p$ does not necessarily impose a bound on the depth at which p occurs in the subtree rooted at s.

   (b) Show that $\mathsf{A}\Diamond\Box\neg p$ is equivalent to the μ-calculus formula $\mu X.\nu Y.[a]((X \vee \neg p) \wedge Y)$.

   (c) Show that $\mathsf{A}\Diamond\Box\neg p$ is not equivalent to any formula of the form $\mu X.\varphi$ or $\nu X.\varphi$, where φ is quantifier-free. (*Hint.* Argue that in every binary-branching tree model, the closure ordinal of any such formula is at most ω.)