CS 410/510spec Takehome Midterm – Due in class Tuesday, Feb. 16

Please spend no more than three hours on this exam, which should be more than enough. You may use any books and notes referenced in class, but please do not consult the internet (not that it would be likely to do you too much good).

Please turn the exam in on paper; there’s no need to type-set it unless your handwriting is hopelessly illegible.

If you have questions, feel free to email me (tolmach@pdx.edu).

Below, you will find three scenarios, each stating some facts and drawing a conclusion, which might or might not be intuitively valid. For each scenario, your task is to give a formal statement of the scenario and a formal verification or refutation of the conclusion.

For each scenario, you must choose to model it in exactly one of the following forms of logic.

- Propositional logic. In this case, your formal statement of the scenario must define all atomic sentences and give a sequent. A verification must be presented as a complete natural deduction proof of the sequent; a refutation must be in the form of a valuation falsifying the sequent.

- Predicate logic. In this case, the formal statement must define all predicates, functions, and constants, and give a sequent. Again, a verification must be presented as a complete natural deduction proof of the sequent; a refutation must be in the form of a model falsifying the sequent.

- Linear temporal logic. In this case, the formal statement must define all atoms, and give a single (implication) formula relating the premises to the conclusion. A verification must be presented as a reduction of that formula to True using equivalence laws from Lamport or Huth&Ryan; a refutation must be in the form of a state sequence that does not satisfy the formula.

Choose the logic that seems to you to best allow a clear statement and analysis of the scenario, without introducing unnecessary detail or complications. There may not be a single “correct” choice in all cases.

Notes:

1. The formal statements are just as important as the verifications/refutations.

2. The facts need not necessarily be in one-to-one correspondence with formal premises.

3. If you feel the need to make further assumptions about a scenario, write them down explicitly.

Turn over...
Scenario 1

Facts:

- All programs have bugs.
- Chris is a programmer who never gives up.
- Programmers always keep writing programs.
- People always test at least some of the programs they write.
- If anyone tests a program that has bugs, they will eventually find one of those bugs or they will stop trying.

Conclusion: Chris will find a program bug some day.

Scenario 2

Facts:

- No safe programs are unpopular on the internet.
- No program in C is free from memory access violations.
- All useful programs operate on null-terminated strings.
- No program with memory access violations is popular on the internet.
- Only C programs operate on null-terminated strings.

Conclusion: No useful program is safe.

Scenario 3

Suppose elections for a certain government office are held each year.

Facts:

- At some point, my family will get interested in politics, and after that, at least one at us will vote in every election from then on.
- Each election ballot consists of one Democratic and one Republican candidate (write-in candidates are not allowed), and one of them always wins.
- Neither party stays in office indefinitely.

Conclusion: eventually someone in my family will vote for the candidate who wins (that year).