CS 330 Formal Methods and Models

Midterm (Fall 2012)

Instructor: Carlotta Domeniconi

October 11, 2012

Student's name:

This test is governed by the GMU Honor Code. The paper you turn in must be your sole work. Help may be obtained from the instructor to understand the description of the problem, but the solution must be the student's own work. Any deviation from this is considered a Honor Code violation.

1. [20 points]

For each of the following 5 questions, NO justification is needed. Just provide your answer.
Note: **T**= True; **F**= False

(a)

$(p \to (q \vee r)) \equiv ((p \wedge \neg q) \to r)$ **T** or **F**

(b)

Suppose that p and q are statements so that $(p \to q)$ is FALSE. Then $(q \to p)$ is TRUE.

T or **F**

(c)

The following argument is valid:

$$\begin{array}{l} p \to q \\ \neg p \\ \hline \neg q \end{array}$$

T or **F**

(d)

$\forall x \in \mathbb{R} : \forall y \in \mathbb{R} : \exists z \in \mathbb{R} : x = y + z$ **T** or **F**

(e)

The following code implements the *negation as failure* in Prolog:

```
not(G).
not(G) :- G, !, fail.
```

T or **F**

2. [20 points]

Using rules of inference with no substitution, prove the following conditional statement.

$$(\neg q \to \neg p) \to (p \to q)$$

Indent your steps as necessary, and label each step with the rule of inference that justifies it. [Note: do **NOT** use the contrapositive equivalence!]

3. [20 points]

Prove the following statement by mathematical induction. [*To receive credit, you must follow the given instructions.*]

$$\sum_{i=0}^{n}(3 \times 5^i) = \frac{3(5^{n+1} - 1)}{4}$$

for all integers $n \geq 0$.

Base case: clearly state what you need to prove as base case, and then prove it.

Inductive step: In this step you must prove $P(n) \rightarrow P(n+1)$, for all n in the proper range. Clearly write the inductive hypothesis and the inductive conclusion. Then proceed with the proof.

Inductive hypothesis: (Specify $P(n)$, and the proper range for n, here)

Inductive conclusion: (Specify $P(n+1)$ here)

Proof:

4. [20 points]

Consider the following pseudo-code. Assume that, at the beginning of its execution, c is a nonzero number which has already been read from the input.

$prod \leftarrow 1;$
$i \leftarrow 0;$
while $(i < n)$ do
    $i \leftarrow i + 1$
    $prod \leftarrow prod * c$

(1) State the loop invariant.

(2) Prove the loop invariant.

(3) Apply the loop invariant.

5. [20 points]

Consider the following Prolog program. It contains facts regarding instructors of classes and in which classes students are enrolled. *instructor*(p,c) means that professor p is the instructor of course c. *enrolled*(s, c) means that student s is enrolled in course c. We want to use these facts to answer queries concerning the professors who teach particular students.

Write a Prolog rule for the predicate *teaches*(P, S) using the predicates *instructor* and *enrolled*. We want *teaches*(p, s) to be true when professor p teaches student s. Add your rule at the end of the given program.

instructor(fibonacci, math100).
instructor(turing, cs330).
instructor(galileo, phys210).
enrolled(john, math100).
enrolled(sofia, cs330).
enrolled(ryan, phys210).
enrolled(lisa, math100).
enrolled(matt, cs330).
enrolled(lisa, cs330).

What would Prolog return given the following queries? If a query has more than one answer, list all the answers.

?- *teaches*(X, lisa).

?- *teaches*(turing, Y).

?- *instructor*(galileo, _).

?- *instructor*(_, ee100).