Name: .....
Student Number .....

Software Engineering 2F04

DAY CLASS — Dr. Mark Lawford
DURATION OF EXAMINATION: 3 Hours
McMaster University Final Examination — December 2001

THIS EXAMINATION PAPER INCLUDES 5 PAGES AND 5 QUESTIONS. YOU ARE RESPONSIBLE FOR ENSURING THAT YOUR COPY OF THE PAPER IS COMPLETE. BRING ANY DISCREPANCY TO THE ATTENTION OF YOUR INVIGILATOR.

Special Instructions: The use of calculators, notes, and text books is not permitted during this exam. Answer all questions in the provided answer booklets. Fill in your name and student number and sign each booklet you use. This paper must be returned with your answers.

NOTE: Tables of proof rules appear at the back of the exam.

1. Predicate Logic Semantics (15 marks)

Consider the following sequence of formulas:

$$\begin{aligned}\Gamma \;:=\; & \exists x(f(x) = x \wedge \forall y(y = f(y) \rightarrow y = x)),\\ & \exists x \exists y(x \neq y \wedge (f(x) = f(y))),\\ & \forall x \exists y(x = f(y))\end{aligned}$$

The intended interpretation for these formulas is given by the model $\mathcal{M}$ with:

$$A \;:=\; \mathbb{N} = \{0, 1, 2, \ldots\} \text{ is the universe}$$

and the interpretation of f, denoted $f^{\mathcal{M}}$, is yet to be determined.

You are given Γ as a formal specification and told to create two different versions of $f^{\mathcal{M}}$.

a) (5 marks) Find an $f^{\mathcal{M}}$ such that for this interpretation of f, $\mathcal{M}$ is a model for the first two formulas in Γ but not the third formula.

b) (7 marks) Next, find a different $f^{\mathcal{M}}$ such that for this interpretation of f we have $\mathcal{M} \models \Gamma$.

c) (3 marks) What does your answer to the previous question tell you about the consistency of Γ?

2. Predicate Logic Proofs & Models (30 marks)

a) (10 marks) Use a formal proof to show that:

$$\forall x(P(x) \rightarrow \neg Q(x)), \forall x(Q(x) \vee \neg R(x)), \exists x \neg Q(f(x)) \vee \exists x Q(f(x)) \vdash \exists x(\neg P(f(x)) \vee \neg R(f(x)))$$

b) (10 marks) Are the premises

$$\Gamma := \forall x(P(x) \rightarrow \neg Q(x)), \forall x(Q(x) \vee \neg R(x)), \exists x \neg Q(f(x)), \exists x Q(f(x))$$

consistent or inconsistent. Justify your answer in the form of a proof or a model.

Continued on page 2

c) (10 marks) Are the premises

$$\Gamma := P(a) \vee \forall x Q(x), \neg P(b), a = b \wedge \neg Q(c)$$

consistent or inconsistent? Justify your answer in the form of a proof or a model.

3. Mathematical Induction (15 marks)

a) (10 marks) Prove using mathematical induction that for all $n \geq 0$, the value of $n^2 + 5n + 1$ is odd.

b) (5 marks) Let f be a unary function symbol that we will interpret as$f^{\mathcal{M}}(x) = x^2 + 5x + 1$. Assuming that our universe of interpretation is the natural numbers, write down a formal predicate logic formula for the statement that we proved in part (a) above.

4. Partial Functions (10 marks)

a) (4 marks) Create the "best" PVS definitions for the function: $f(x, y) = \ln(x - y^2)$

b) (6 marks) Write down the most concise formulas in both the IMPS/Parnas (analysis style) logic and bounded quantification (PVS style) logic that could be used to specify that A, an N element array of integers, has the property:

The array does not contain a strictly increasing sequence of elements.

5. Software Verification (30 marks)

Consider the code fragment shown in Figure 1 that is supposed to implement a linear search for an element `key` in an n element array that runs from 0 to $n - 1$.

In order to check that this code fragment is correct, a diligent program has formulated the PVS input shown in Figure 2 to verify the correctness of the loop. (NOTE: `(A(i)/=key)` is short for `NOT(A(i)=key)`.) The programmer tries to prove all of the TCCs generated by the file and the `THEOREM`s `C1`, `C2` and `C3`. It is possible to prove `C1` and `C3` but not `C2`. Also, there is one TCC the programmer is unable to prove.

The TCC that the programmer is unable to prove is shown in Figure 3 together with the sequent that results when trying to prove it.

a) (5 marks) If the programmer had been able to prove all of the theorems, could the programmer ignore the unproven TCC? Explain your answer briefly with reference to the purpose of TCCs in PVS.

b) (5 marks) Write down the characteristic equation for the above sequent and find a counter example to the equation.

c) (5 marks) Is your counter example for the sequent a counter example for `B_TCC1` proof obligation?

d) (5 marks) Why does the definition of predicate `B` generate `B_TCC1` and why is it unprovable?

e) (5 marks) The predicate `B(i,key)` is taken from the condition of the **while**. What is this unprovable TCC telling you is wrong with the program fragment? How could the program fragment be fixed?

f) (5 marks) The predicates `I(i,key)` and `P(i,key)` both make use of the expression:

```
(forall (j:{k:nat|k<=i-1}): A(j)/=key)
```

Continued on page 3

$i := 0$;
while $0 \leq i \leq n \wedge A(i) \neq key$ **do**
    $i := i + 1$;
end;

Figure 1: Program fragment for Question 5

```
prog_ver : THEORY
  BEGIN

  n: int
  i: VAR nat
  AType: TYPE+
  key: VAR AType

  A(i:{k:nat|k<=n-1}):AType

  V(i,key):bool = (0<= n)
  B(i,key):bool = (0<=i) & (i<=n) & (A(i)/=key)
  I(i,key):bool = (0<=i) & (i<=n) & (forall (j:{k:nat|k<=i-1}): A(j)/=key)
  P(i,key):bool = (0<=i) & (i<=n) &  (forall (j:{k:nat|k<=i-1}): A(j)/=key)
                   & (i=n OR A(i)=key)

  C1: THEOREM V(i,key) => I(0,key)
  C2: THEOREM I(i,key)& B(i,key) => I(i+1,key)
  C3: THEOREM I(i,key)& NOT B(i,key) => P(i,key)

  END prog_ver
```

Figure 2: PVS code for Question 5

```
% Subtype TCC generated by B(i,key) for i
  % unfinished
B_TCC1: OBLIGATION
  FORALL (i): 0 <= i AND i <= n IMPLIES i >= 0 AND i <= n - 1


[-1]  0 <= i!1
[-2]  i!1 <= n
  |-------
[1]   i!1 <= n - 1

Rule?
```

Figure 3: Unproven TCC for Question 5

Continued on page 4

What is the value of this expression when $i = 0$? Does it depend upon the interpretation of the array `A` and the value of `key`?
(HINT: Consider the negation of the formula.)

"What are my options if I want to transfer out of software? " - 2F04 student

Propositional Logic Proof Rules

	introduction	elimination
$\wedge$	$\dfrac{\phi \quad \psi}{\phi \wedge \psi} \wedge i$	$\dfrac{\phi \wedge \psi}{\phi} \wedge e_1 \quad \dfrac{\phi \wedge \psi}{\psi} \wedge e_2$
$\vee$	$\dfrac{\phi}{\phi \vee \psi} \vee i_1 \quad \dfrac{\psi}{\phi \vee \psi} \vee i_2$	$\dfrac{\phi \vee \psi \quad \begin{array}{|c|}\hline \phi \\ \vdots \\ \chi \\ \hline \end{array} \quad \begin{array}{|c|}\hline \psi \\ \vdots \\ \chi \\ \hline \end{array}}{\chi} \vee e$
$\rightarrow$	$\dfrac{\phi \quad \phi \rightarrow \psi}{\psi} \rightarrow e$	$\dfrac{\begin{array}{|c|}\hline \phi \\ \vdots \\ \psi \\ \hline \end{array}}{\phi \rightarrow \psi} \rightarrow i$
$\leftrightarrow$	$\dfrac{\phi \rightarrow \psi \quad \psi \rightarrow \phi}{\phi \leftrightarrow \psi} \leftrightarrow i$	$\dfrac{\phi \leftrightarrow \psi}{\phi \rightarrow \psi} \leftrightarrow e_1 \quad \dfrac{\phi \leftrightarrow \psi}{\psi \rightarrow \phi} \leftrightarrow e_2$
$\neg$	$\dfrac{\begin{array}{|c|}\hline \phi \\ \vdots \\ \bot \\ \hline \end{array}}{\neg\phi} \neg i$	$\dfrac{\phi \quad \neg\phi}{\bot} \neg e$
$\neg\neg$	$\dfrac{\phi}{\neg\neg\phi} \neg\neg i$	$\dfrac{\neg\neg\phi}{\phi} \neg\neg e$
$\bot$	see $\neg e$	$\dfrac{\bot}{\phi} \bot e$

Continued on page 5

Additional Propositional Logic Proof Rules

$$\frac{\phi \to \psi}{\neg\phi \vee \psi} \to 2\vee \qquad \frac{\neg\phi \vee \psi}{\phi \to \psi} \vee 2 \to$$

$$\frac{\phi \to \psi \quad \neg\psi}{\neg\phi}\text{MT} \qquad \frac{\boxed{\begin{array}{c}\neg\phi \\ \vdots \\ \bot\end{array}}}{\phi}\text{RAA}$$

$$\frac{}{\phi \vee \neg\phi}\text{LEM}$$

Additional Predicate Logic Proof Rules

	introduction	elmination
$\forall$	$\dfrac{\boxed{\begin{array}{l}x_0 \\ \quad\vdots \\ \quad\phi[x_0/x]\end{array}}}{\forall x\phi}\forall i$	$\dfrac{\forall x\phi}{\phi[t/x]}\forall e$
$\exists$	$\dfrac{\phi[t/x]}{\exists x\phi}\exists i$	$\dfrac{\exists x\phi \qquad \boxed{\begin{array}{ll}x_0 & \phi[x_0/x] \\ & \vdots \\ & \chi\end{array}}}{\chi}\exists e$
$=$	$\dfrac{}{t = t} = i$	$\dfrac{t_1 = t_2 \quad \phi[t_1/x]}{\phi[t_2/x]} = e$

The End