HAND IN EXAM
Answers recorded on examination paper

Queen's University
Faculty of Arts and Science
School of Computing
CISC422
Final Examination

Wednesday, December 8, 2010

Instructor: J. Dingel

Student id (clearly printed please): ______________________________

Instructions: This examination is **three hours** in length. You are permitted to have one 8.5"x11" data sheet with information on both sides. No other references, calculators, or aids are allowed.

Read all questions carefully. There are 5 questions in this examination, some with several parts. You must answer all questions. Write all your answers in this exam. An extra page is provided at the end for rough work. If you answer a question on a different page, you must indicate where the answer is to be found.

Important notes: Make sure your student id is clearly printed on this page. If the instructor is unavailable in the examination room and doubt exists as to the interpretation of any question, you are urged to submit your answer with a clear statement of any assumptions made.

Good luck!

For marking use only:

Question	Points
1	/16
2	/37
3	/18
4	/16
5	/38
Total	/125

Id: ____________________

Question 1: (Predicate Logic, 16 points)

Consider the following sets of variables and symbols:

$$\mathcal{V} = \{p, q, e\} \qquad \mathcal{F} = \{f, g\} \qquad \mathcal{P} = \{P, Q\}$$

where the variables p and q range over *"source programs"* (e.g., .java or .c files) and e range over *"executables"* (e.g., .class or .exe files), and the function symbols f and g (both with arity 1), and the predicate symbols P (with arity 1) and Q (with arity 2) have the following meaning:

- $f(p)$: *"returns the executable that results from compiling a syntactially correct source program p"*
- $g(p)$: *"returns the source program that results from modifying source program p"*
- $P(p)$: *"the source program p is syntactically correct"*
- $Q(p, e)$: *"the source program p and the executable e are 'in sync', i.e., p is syntactially correct and compiling p yields e"*

Using the above symbols, formalize the following statements in Predicate Logic:

1) (4 pts) *"The compilation of a syntactically correct source program results in an executable that is in sync with the source program"*

2) (4 pts) *"The modification of a source program p will never be in sync with the executable of p"*

3) (4 pts) *"Equality is preserved by compilation, i.e., equal source programs will give rise to equal executables"*

4) (4 pts) *"If modifying some program p makes it syntactically incorrect, then the modified program will not be in sync with the executable resulting from the compilation of p"*

Id: ____________________

Question 2: (Alloy, 37 points)

Answer the following questions below. Each answer should be brief (i.e., 2-3 sentences), yet as clear and precise as possible.

1. (3 pts) What is the satisfiability problem for Alloy formulas?

2. (3 pts) The satisfiability problem for Alloy formulas is not decidable. What does that mean?

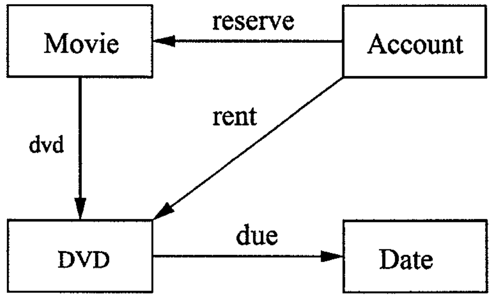
3. (4 pts) The Alloy Analyzer, essentially, reduces every analysis to a satisfiability problem. Which technique ("trick") does the Alloy Analyzer use to deal with the undecidability of the satisfiability problem?

4. (4 pts) Which consequences does this have, i.e., what are the advantages and disadvantages of this technique?

Id: ____________________

Question 2: (Alloy, continued)

Consider the following class diagram for the management system of a video store



where the classes have the following informal description:

Movie Instances of this class are all movies that have been released in the theaters or on video to date.

DVD Instances of this class are the DVDs that the customers take home. Possible attributes of a DVD include the title of the movie stored on it, its age, and, if it is rented, the due date.

Account Instances of this class stand for individual customers. Possible attributes of an account include the name of the customer, the DVDs currently rented by the customer, the movies currently reserved by the customer, the telephone number etc.

Date Instances of this class are calendar dates.

The meaning of the relations should be self-explanatory.

1. (7 pts) Express the class diagram as a textual Alloy specification. Moreover, using the informal description above and your common sense, augment your textual Alloy specification with the appropriate multiplicity constraints. Note that you don't need to add any of the attributes mentioned in the informal description above.

Id: ____________________

Question 2: (Alloy, continued)

2. Using your Alloy specification from the previous page, express each of the following informal invariants formally as facts in Alloy.

   1) (4 pts) *"For all DVDs, there exists exactly one movie that is stored on it".*

   2) (4 pts) *"A movie can only be reserved, if there is a DVD for it".*

   3) (4 pts) *"A customer can only reserve a movie if she is not currently renting a DVD of it".*

   4) (4 pts) *"A DVD has a due date if and only if it is rented".*

Id: ____________________

Question 3: (CTL, 18 points)

For each of the following formulas φ,
a) draw a non-trivial Kripke structure (i.e., finite state machine) M such that φ holds in M, and
b) draw a non-trivial Kripke structure such that φ does not hold in M.
"Non-trivial" here means that that your state machine must have at least 3 states. Clearly indicate the initial state and which atomic propositions hold in which states.

1. (6 pts) $(\neg p) \wedge (\mathbf{EF}\ p) \wedge (\mathbf{EG}\ q)$

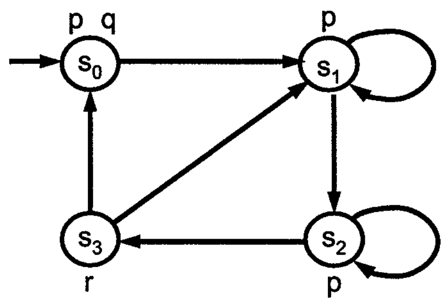
2. (6 pts) $(\mathbf{EG}\ \neg q) \wedge \mathbf{E}[p\ \mathbf{U}\ (\mathbf{AF}\ q)]$

3. (6 pts) $(\neg\mathbf{EG}\ \neg p) \wedge \mathbf{AG}\big(p \rightarrow (\neg q \wedge \mathbf{AF}\ q)\big)$

Id: ____________________

Question 4: (Model checking, 16 points)

Consider the following graphical representation of a Kripke structure (i.e., finite state machine) M.



1. (4 pts) Define M textually as a 4-tuple $M = (S, S_0, R, L)$.

2. (4 pts) Beginning from state s_0, unwind M into its corresponding computation tree T. Draw T to a depth of 4. More precisely, you must show all computation paths of M starting at s_0 up to length 4. The length of a path is the number of edges on it.

Id: ____________________

Question 4: (Model checking, continued)

4. (8 pts) For each of the following CTL formulas φ decide whether the formula holds in machine M defined on the previous page. If your answer is **"No"**, that is, φ does not hold in M, then give a counter example, that is, an execution path in M illustrating the **violation** of φ. Remember that paths are always infinite. Use periods "..." and parentheses to indicate that a sequence of states is repeated infinitely often.

   (a) $\mathbf{AF}\ \mathbf{EG}p$

   (b) $\mathbf{AG}(q \rightarrow \mathbf{AG}p)$

   (c) $\mathbf{AG}((q \wedge \mathbf{AX}p) \rightarrow \mathbf{EG}p)$

   (d) $\mathbf{AG}\ \mathbf{AF}r$

Id: ____________________

Question 5: (SMV, 38 pts)

Consider the following SMV code.

```
MODULE P1
VAR
  x :  {c, h, t};
ASSIGN
  init(x) := c;
  next(x) := case
                 x=c :  {h, t};
                 TRUE: c;
              esac;
```

```
MODULE P2(x)
VAR
  y :  {0, 1, 2};
ASSIGN
  init(y) := 2;
  next(y) := case
                   x=h & y=2 :  1;
                   x=h & y=1 :  0;
                   x=h & y=0 :  2;
                   TRUE : y;
               esac;
```

```
MODULE main
VAR
  p1 :  P1;
  p2 :  P2(p1.x);
```

1. (9 pts) Draw the Kripke structure (i.e., finite state machine) M that is defined by the code above. Represent a single state s of M by a pair (x, y) where x is the value of x in process p1 and y is the value of y in process p2. For instance, the initial state of M is represented as $(c, 2)$. Your drawing should clearly indicate the initial states of M, the reachable states of M, and the transition relation of M. You don't need to show the labelling function.

Id: ____________________

Question 5: (SMV, continued)

2. (4 pts) Beginning from the initial state of the machine M (from the previous page), unwind M into its corresponding computation tree T. Draw T to a depth of 4. More precisely, you must show all computation paths of M starting at the initial state up to length 4. The length of a path is the number of edges on it.

Id: ____________________

Question 5: (SMV, continued)

3. (25 pts) For each of the following properties φ_1 through φ_5, express it formally in CTL and decide whether or not it is true in your finite state machine M given in Part 1 of this question by writing "Yes" or "No". No justification necessary. In each of the statements below, we use `p1.x` to refer to the value of variable `x` in process `p1`. Similarly for `p2.y`.

   1. (5 pts) φ_1: "Along every path it is always the case that if `p1.x` is `h` and `p2.y` is 2, then `p2.y` is 1 in all next states".
      - φ_1 in CTL:

      - φ_1 true in M?:
   2. (5 pts) φ_2: "For all states s along every path, if `p1.x` is `h` in s and `p2.y` is 2 in s and `p2.y` is 1 in all successor states of the successor states of s, then `p1.x` is `h` in all successor states of s".
      - φ_2 in CTL:

      - φ_2 true in M?:
   3. (5 pts) φ_3: "Along every path, `p1.x` will be `h` eventually and `p2.y` will be 2 until then".
      - φ_3 in CTL:

      - φ_3 true in M?:
   4. (5 pts) φ_4: "There exists a path along which `p2.y` is always 2".
      - φ_4 in CTL:

      - φ_4 true in M?:
   5. (5 pts) φ_5: "In every state, along every path, if `p2.y` is 2, then `p2.y` is 2 or 0 in all successor states".
      - φ_5 in CTL:

      - φ_5 true in M?:

Id: ______________________

Scratch sheet: