UNIVERSITY OF TORONTO
Faculty of Applied Science and Engineering
Final Examination in ECE450S Software Engineering II
2001 April 24, 2pm-4:30pm. Duration: 2.5 hours
No Aids. A list of algebraic laws is provided separately.

FAMILY NAME:________________________

GIVEN NAMES:________________________

STUDENT NUMBER:______________________

The value of each question is indicated in square brackets, total [150]. A blank answer is worth about one-third of the marks; to that, marks will be added for correct and relevant information, and marks will be subtracted for incorrect or irrelevant information. Write neatly; unreadable answers are worthless.

Answer each question in the space provided. Use the backs for rough work. There are 150 marks total and 150 minutes for the exam.

1.	/27	6.	/18
2.	/ 9	7.	/ 9
3.	/15	8.	/18
4.	/12	9.	/18
5.	/15	10.	/ 9

1 Answer each of the following questions briefly.

(a)[3] Define "formal specification". By observing something, how can you tell if it satisfies a formal specification?

(b)[3] Let S be a specification in memory variables x and y and time variable t. When is S implementable?

(c)[3] Let A and B be specifications in memory variables x and y and time variable t. When is specification A refined by specification B?

(d)[3] What is the difference between an architectural description and a behavioral description of the same system?

(e)[3] When are BDDs not the best symbolic data structure, and why not?

(f)[6] Write an SMV program in which the two fairness constraints

```
FAIRNESS a
FAIRNESS b
```

are not equivalent to the one fairness constraint

```
FAIRNESS a & b
```

(g)[3] Let x be an integer state variable. Is $\neg(x \geq 0 \ \wedge \ x' = 0)$ implementable? Why or why not?

(h)[3] Let x be an integer state variable. Is $\neg(x \geq 0 \ \vee \ x' = 0)$ implementable? Why or why not?

2[9] Give an architectural description diagram in Darwin using symbolic (not textual) description for the following UNIX command:

```
find . –name "straightA" | grep 450 | wc
```

find computes locations of files named "straightA" and gives their path relative to . (dot). grep searches files for lines containing the string "450" and outputs those lines. wc reads input files and writes the number of newline characters, words and bytes.

3 Express each of the following using an appropriate formalism from the course.

(a)[3] In some computation action a is eventually followed by action b.

(b)[3] Exactly one of a, b, or c is true.

(c)[3] Neither your money nor my money is enough, but your money and my money is enough.

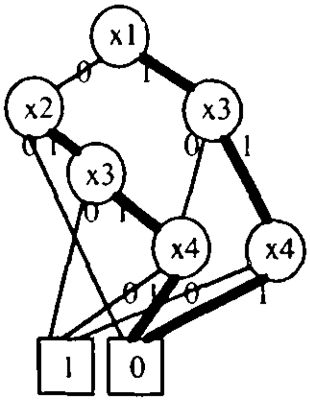
(d)[6] (one elevator, two floors) The elevator keeps its door open until there is a request to go to the other floor, and even then it keeps its door open if there's a request to stay where it is.

4 Which of the following pairs of CTL formulas are equivalent? For those that are, prove that they are, using the CTL definitions and laws. For those that are not, exhibit a model of one of the pair that is not a model of the other.

(a)[6] $\mathrm{EF}\phi \vee \mathrm{EF}\psi$ and $\mathrm{EF}(\phi \vee \psi)$

(b)[6] $\mathrm{A}[\phi \mathbin{\mathrm{U}} \mathrm{A}[\psi \mathbin{\mathrm{U}} \pi]]$ and $\mathrm{A}[\mathrm{A}[\phi \mathbin{\mathrm{U}} \psi] \mathbin{\mathrm{U}} \pi]$

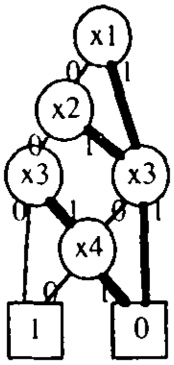
5 Here is a BDD.



(a)[9] What expression does it indicate, in conjunctive normal form (a conjunction of disjunctions)?

(b)[3] Is it a ROBDD? If not, please make it one.

(c)[3] Does the following ROBDD indicate the same expression?



6[18] Let x and n be natural variables (that means a variable whose value is a natural number). Find a specification P such that both the following are theorems, and prove them.

$$x = x' \times 2^{n'} \quad \Leftarrow \quad n := 0;\ P$$

$$P \quad \Leftarrow \quad \textbf{if}\ \mathit{even}\ x\ \textbf{then}\ (x := x/2;\ \ n := n+1;\ \ P)\ \textbf{else}\ \mathit{ok}$$

(where the integer division rounds down)

7[9] Let n and r be natural variables (that means a variable whose value is a natural number) in the refinement

$$P \Leftarrow \textbf{if } n=1 \textbf{ then } r:=0 \textbf{ else } (n:=n/2;\ P;\ r:=r+1)$$

(where the integer division rounds down). Suppose the operations / and + each take time 1 and all else is free (even the call is free). Insert appropriate time increments, and find an appropriate P to express the execution time. (You do not need to prove the refinement for your choice of P.)

8[18] A natural number can be written as a sequence of decimal digits with a single leading zero. Using the notation and methods of this course, given two natural numbers, write a program to find the number that is written as their longest common prefix of digits. For example, given 025621 and 02547, the result is 025. State all specifications, but you do not need to prove the refinements. Hint: this question is about numbers, not about strings or lists.

9 You are given the following SMV model.

```
MODULE one(a, b)
     ASSIGN
        next(a):= case b & !a: 1;
                       1: a;
                  esac;
MODULE two(a, b)
     ASSIGN
        next(b):= case a: 1;
                       1: {0, 1};
                  esac;
MODULE main
     VAR
        a, b: boolean;
        run_one: process one(a, b);
        run_two: process two(a, b);
     ASSIGN
        init(a):= 0;
        init(b):= 1;
```

(a)[1] Is the parallelism synchronous or asynchronous?

(b)[3] Which states can the system be in after one step?

(c)[9] Draw a finite state machine corresponding to the SMV model.

(d)[5] Show, step by step, how the model checker checks the formula $E[\neg a \ U \ \neg b]$. Is this property true?

10[9] We are specifying a thermostat system that keeps the temperature in a room in a comfort range. It has a switch *running* with which we can turn the system on or off. The temperature can be *belowDesired* , *aboveDesired* , or *Desired* . The system can run an air conditioner or heater. Here is a model of the controller.

```
MODULE main
  aboveDesired, belowDesired, Desired, running: boolean;
  Device: {Idle, Heat, AC};
  ASSIGN
    init(running):= 1;
    next(running):= case aboveDesired: 1;
                         belowDesired: 1;
                         Desired: 0;
                    esac;
    init(Device):= Idle;
    next(Device):= case aboveDesired: AC;
                        belowDesired: Heat;
                        Desired: Idle;
                   esac;
```

What is wrong with this model? Please correct it.