Exam No

The University of British Columbia

Computer Science 121

Midterm 2

March 15, 2012

Time: 70 minutes
Total marks: 60

Instructors (Circle one): Section 202 - Patrice Belleville

Section 203 & BCS - George Tsiknis

Name____Student No_______________
(PRINT) (Last) (First)

Signature____

This examination has 8 pages.

Check that you have a complete paper.

This is a closed book exam, but you may use a sheet of 8.5x11 inches paper with your notes.

Answer all the questions on this paper.

Give very **short but precise** answers. Always use point form where it is appropriate.

Work fast and do the easy questions first. Leave some time to review your exam at the end.

The marks for each question are given in []. Use this to manage your time.

Good Luck

M A R K S	
1	
2	
3	
4	
5	
6	
Total	

<u>Question 1.</u> [10] Formal Proofs in Predicate Logic

a. [4] Prove that :

$\sim\forall x\in D, \exists y\in D, (P(x, y) \land \sim Q(x, y)) \lor (P(x, y) \land Q(x, y)) \equiv \exists x\in D, \forall y\in D, \sim P(x, y)$

<u>Proof:</u>

$\sim\forall x\in D, \exists y\in D, (P(x, y) \land \sim Q(x, y)) \lor (P(x, y) \land Q(x, y))$	
$\equiv \exists x\in D, \forall y\in D, \sim((P(x, y) \land \sim Q(x, y)) \lor (P(x, y) \land Q(x, y)))$	**quantifier neg.**
$\equiv \exists x\in D, \forall y\in D, \sim(P(x, y) \land (\sim Q(x, y) \lor Q(x, y))$	**absorption**
$\equiv \exists x\in D, \forall y\in D, \sim(P(x, y) \land T)$	**idempotent**
$\equiv \exists x\in D, \forall y\in D, \sim P(x, y)$	**identity**

b. [6] Provide a **formal proof** (by using one formal rule of deduction at a time and explicitly stating the rule that is used on the right side) for the following argument. You may assume that **a∈ D, b∈ D** and **c∈ D**:

$\forall x\in D, \forall y\in D, (\exists z\in D, W(x, z) \land W(y, z)) \rightarrow x=y$	**(1)**
<u>$W(a,b)$</u>	**(2)**
$a=c \lor \sim W(c,b)$	

<u>Proof:</u>

3. $\forall y\in D, (\exists z\in D, W(a, z) \land W(y, z) \rightarrow a=y)$	**(1), Univ. Instantiation**
4. $(\exists z\in D, W(a, z) \land W(c, z)) \rightarrow a=c$	**(1), Univ. Instantiation**
5. $\sim(\exists z\in D, W(a, z) \land W(c, z)) \lor a=c$	**def of →**
6. $\forall z\in D, \sim(W(a, z) \land W(c, z)) \lor a=c$	**quantifier negation**
7. $\sim(W(a, b) \land W(c, b)) \lor a=c$	**Univ. Instantiation**
8. $\sim W(a, b) \lor \sim W(c, b) \lor a=c$	**de Morgan**
9. $\sim W(c, b) \lor a=c$	**(7), (2) elimination**

Question 2. [8] Proof Techniques I

Express each of the following statements in predicate logic, and then suggest an appropriate proof strategy for it (pick one of the techniques we have seen in the class: witness proof, generalizing from a generic particular, proof by cases, contrapositive proof or proof by contradiction). Note that you **must NOT prove** the statement. You just need to suggest a proof strategy which is **the most substantial part** of the proof of the statement..

a. [2] For any integer n if n *mod* 5 is 3 than n^2 *mod* 5 is 4. Recall that x *mod* y is the remainder when x is divided by y.

Formula:

$\forall n \in Z, n \bmod 5 = 3 \rightarrow n^2 \bmod 5 = 4$

Proof strategy:

proof by generalizing from a generic particular (or direct proof)

b. [2] If the product of two positive real numbers is less than 10000, at least one of them is less than 100. Use R^+ to represent the set of all positive real numbers.

Formula:

$\forall x \in R^+, \forall y \in R^+, \ x \cdot y < 100 \rightarrow ((x < 100) \vee (y < 100))$

Proof strategy:

proof by contrapositive or proof by contradiction

c. [2] Not every animal that has wings flies (use A for the domain of all animals, W(x) ≡ x has wings and F(x) ≡ x flies).

Formula:

$\sim\forall a \in A, W(a) \rightarrow F(a)$

Proof strategy:

witness proof

d. [2] For every integer n, n^2 is odd if and only if n is odd (use predicates Odd(x) and Even(x) to indicate odd and even integers) .

Formula:

$\forall n \in Z, Odd(n^2) \leftrightarrow Odd(n)$

Proof strategy:

proof by cases or

proof by contrapositive

Question 3. [8] Proof Techniques II: Complete the Proof

The following is an incomplete proof for the statement

`for any integer n, n² – 2 is not divisible by 4`

We first express the statement in Predicate Logic using the predicate

`Divisible(x,y)` ≡ x is the product of y and another integer k.

Some parts of the proof are left out and are shown by "______". Fill in these parts to complete this proof.

Proof:

Expressed in Propositional Logic, the statement that we need to prove is:

$\forall n \in \mathbb{Z}$, ~Divisible(n^2-2, 4)

Let n be any non specific integer. We need to show

~Divisible(n^2-2, 4)

Assume that

Divisible(n^2-2, 4) (1)

is true and we'll derive a contradiction.

From (1)

$n^2-2 = 4k$ for some integer k

or $n^2 = 4k+2$ (2)

Since n^2 is even, n must be even by the relevant theorem we have proved in the class.

Let **n** = $2m$ for some integer m.

From (2) we get

$4m^2 = 4k+2$

or

$2m^2 = 2k+1$

But the left hand side of this equation is even and the right hand side is odd. This is a contradiction.

Therefore we can conclude `~Divisible(` n^2`—2, 4).`

Question 4. [10] Proof Techniques III

Provide a proof for the following statement :

`For all integers m and n, if m+n is even then m and n are both even or m and n are both odd`

You must first express the statement in Predicate Logic and then construct a **proof by contraposition** following the basic steps we discussed in the class. You can use the predicates `Even(x)` and `Odd(x)` to indicate that x is an even or odd number.

The statement in Predicate logic:

$$\forall n \in Z, \forall m \in Z, \text{Even}(n+m) \rightarrow (\text{Even}(n) \wedge \text{Even}(m)) \vee (\text{Odd}(n) \wedge \text{Odd}(m))$$

Proof:

Let n, m be two unspecified integers.

We 'll show that

$$\sim ((\text{Even}(n) \wedge \text{Even}(m)) \vee (\text{Odd}(n) \wedge \text{Odd}(m))) \rightarrow \sim\text{Even}(n+m)$$

or

$$(\sim(\text{Even}(n) \wedge \text{Even}(m)) \wedge \sim(\text{Odd}(n) \wedge \text{Odd}(m))) \rightarrow \sim\text{Even}(n+m)$$

Assume that one of n and m is even and the other is odd. Without loss of generality let n be even and m be odd, Then

$n = 2a$ for some integer a

and

$m = 2b+1$ for some integer b

Then

$m+n = 2a+2b+1 = 2(a+b) + 1$

Since a+b is an integer, n+m is odd, and therefore n+m is not even.

<u>Question 5.</u> [16] Finite State Automata

Consider the following DFA:

a. [4] Which of the following sequences of 0's and 1's are accepted by the DFA? Simply circle one of the choices besides the input sequence:

i.010101 YES [✓ NO]

ii.01111 [✓ YES] NO

iii.0111110 [✓ YES] NO

iv.011110 YES [✓ NO]

b. [4] Using the cases you have seen in (a) provide a short description of the language that is accepted by the automaton.

The DFA accepts the following sequences of 0's and 1's

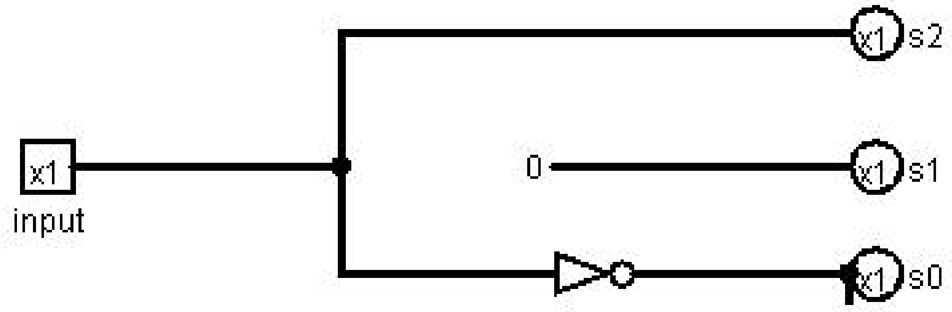
- a single 0
- a single 0 followed by an even number of 1's
- a single 0 followed by an odd number of 1's and a single 0 at the end.

a. [4] The following is the truth table that describes the transition function for the first two states of this automaton. Fill in the part of the table that defines the transitions from states 2 and 3.

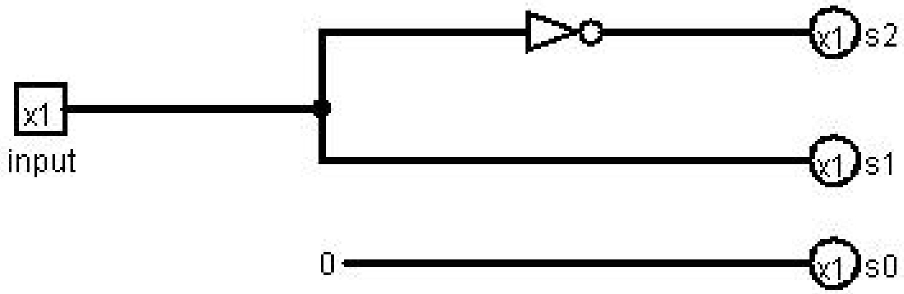
Current State			**Input**	**Next State**		
s2	**s1**	**s0**		**s2**	**s1**	**s0**
0	0	0	0	0	0	1
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	0	0	1
0	1	1	0	1	0	0
0	1	1	1	1	0	0

b. [4] Suppose we implement this automaton in the way we discussed in the class (using flip-flops for the current state, a multiplexer that selects the next state etc.). Provide the circuits that will implement the transitions for state 0 and 1 as they are defined in the above table.

Circuit for state 0:



Circuit for state 1:


x1
input
x1
s2
x1
s1
0
x1
s0

<u>Question 6. [8] Sequential Circuits</u>

In Lab 5 you learned how to use a use a D flip-flop to design a frequency divider and used that idea to develop a counter. Using this idea, complete the circuit that is given below to create a 3-bit counter that counts from 0 to 7 each time the clock ticks (low or high).

Your design CANNOT use any additional clocks or D flip-flops. You may only use wires and simple gates to complete your circuit. Note that for this circuit you don't need to use any of the 3 pins that are on the south side of the flip-flops. Use only the pins that are on their east and west sides.

Make sure that your circuit counts from 0 to 7 and that the output pins `bit0`, `bit1` and `bit2` have the corresponding bits of the counter value.