CPSC 121 Midterm 2
Tuesday, March 19th, 2013

Name: ____________________ Student ID: ____________________

Signature: ____________________ Section (circle one): Morning Afternoon

– You have 70 minutes to write the 7 questions on this examination. A total of 50 marks are available.

– **Justify all of your answers.**

– You are allowed to bring in one hand-written, double-sided 8.5 x 11in sheet of notes, and nothing else.

– Keep your answers short. If you run out of space for a question, you have written too much.

– The number in square brackets to the left of the question number indicates the number of marks allocated for that question. Use these to help you determine how much time you should spend on each question.

– Use the back of the pages for your rough work.

– **Good luck!**

Question	Marks
1	
2	
3	
4	
5	
6	
7	
Total	

UNIVERSITY REGULATIONS:

– Each candidate should be prepared to produce, upon request, his/her UBC card.

– No candidate shall be permitted to enter the examination room after the expiration of one half hour, or to leave during the first half hour of the examination.

– CAUTION: candidates guilty of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action.

1. Having at the place of writing, or making use of, any books, papers or memoranda, electronic equipment, or other memory aid or communication devices, other than those authorised by the examiners.
2. Speaking or communicating with other candidates.
3. Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.

– Candidates must not destroy or mutilate any examination material; must hand in all examination papers; and must not take any examination material from the examination room without permission of the invigilator.

[9] 1. Proof Techniques

[3] a. When we use a direct proof for a statement of the form $\forall x \in D, P(x) \rightarrow Q(x)$, we first consider an unspecified element x of the domain, and then assume that $P(x)$ is true. Why do we make this assumption?

[3] b. One of your friends claims to have proved the theorem

$$\forall n \in \mathbf{Z}^+, \forall x \in \mathbf{Z}^+, \forall y \in \mathbf{Z}^+, \forall z \in \mathbf{Z}^+, P(x, y, z, n)$$

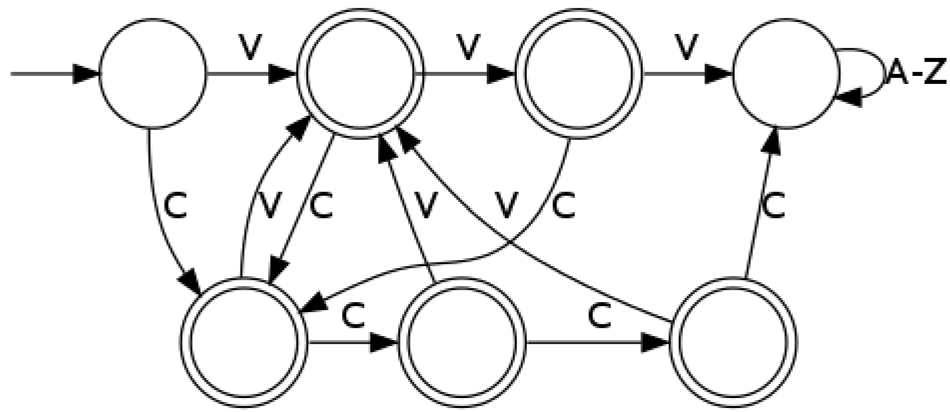
where $P(x, y, z, n)$ is some complicated predicate involving x^n, y^n and z^n. You do not believe him, as you think his "theorem" is false. How would you prove this?

[3] c. In order to prove the theorem

$$\exists x \in \mathbf{Z}^+, \forall y \in \mathbf{Z}^+, Q(x, y)$$

a CPSC 121 instructor from a previous term chooses $x = y^2$ and an unspecified positive integer y, and then proves $Q(y^2, y)$ without making mistakes. Is the intructor's proof valid? Explain why or why not.

[7] 2. Consider the following finite-state automaton, whose inputs are words written using upper-case letters (A through Z) only. The letter "C" on an edge denotes any consonant (B-D,F-H,J-N,P-T,V-Z), whereas the letter "V" denotes any vowel (A,E,I,O,U).



[4] a. Which of the following words will this finite-state automaton accept (circle one of Yes/No for each word)?

• **ACQUAINTANCE**	Yes	No
• **AVADAKEDAVRA**	Yes	No
• **ENTHRONE**	Yes	No
• **FRIENDSHIP**	Yes	No
• **GOOEY**	Yes	No
• **MOTHERHOOD**	Yes	No
• **SCHNITZEL**	Yes	No
• **TROMBONE**	Yes	No

[3] b. Describe as simply as you can the set of words that this finite-state automaton accepts.

[4] 3. Sketch a proof of the following theorem, using a proof technique of your choice:

> We can find integers x and y such that $P(x)$ and $P(y)$ and $P(xy + z)$ for some integer z.

As we did not tell you what property P is, you will not be able to write a complete proof. However write as much of it as you can.

[5] 4. Sketch a proof by contradiction of the following theorem:

> You are in a lab section with a total of X students (including yourself). All X+2 of you (both teaching assistants are also present) write down the day of the week on which your birthdays fall this year. Your instructor then walks in and asks you to prove that at least Y of you wrote down the same day of the week.

As we did not tell you what X and Y are you will not be able to write a complete proof. Write as much of it as you can; stop when you reach a point where you would need to know the values of X and Y.

[5] 5. Sketch a proof by mathematical induction of the following theorem:

> For every positive integer n, the product of the first n numbers with property P is equal to $f(n)$.

As we did not tell you what property P is, or what $f(n)$ is, you will not be able to write a complete proof. However write as much of it as you can.

[11] 6. Consider the following theorem:

> Every odd integer is the sum of two integers whose difference is equal to 5.

We can easily verify that this theorem holds for $n = 19$, for instance, since $19 = 7 + 12$. It also holds for 15, since $15 = 5 + 10$.

[3] a. Translate this theorem into predicate logic (that is, using predicates and quantifiers). The only non-standard predicate you can use is *Odd* (things like $=$ and $\leq$ are allowed, however). Hint: try to only use two variables.

(this question continues on the next page)

[8] b. Prove the theorem using a *direct proof*. Some of the marks for this proof will be given for the structure of your proof; others will be given for the contents.

[9] 7. Recall that Euclid's algorithm to compute the greatest common divisor (GCD) of two positive integer works as follows:

```
(define (gcd a b)
    (if (= b 0)
        a
        (gcd b (remainder a b))))
```

The following is an incomplete sequential circuit to compute the GCD of a pair of positive integers. The computation should start once the initial values of a and b have been entered into the two 8-bit registers `a` and `b`. A number of clock cycles later, the LED labeled `Valid` will be ON, and the output labeled `gcd(a,b)` will contain the GCD of a and b.

Complete the implementation of this circuit. The component labeled "upper" and "rem" is a Divider subcircuit: it takes one 16-bit input x (the 8 bits from the top input, and the 8 bits from the upper-left input) and one 8-bit input y (the 8 bits from the bottom-left input). It produces the quotient x/y (the value obtained by writing `(quotient x y)` in Dr. Racket) on the right output, and the remainder (`(remainder x y)`) on the bottom output.

Hint: consider the correspondence between the parameters in one recursive call and the parameters of the next one.