CPSC 121: Models of Computation
Midterm Exam #1, 2009 October 8

Solution : SAMPLE SOLUTION

Name: ______________________ Student ID: ______________________

Signature: ______________________

- You have **75 minutes** to write the 7 questions on this exam.
- A total of **60 marks** are available. You may want to complete what you consider to be the easiest questions first!
- Ensure that you clearly indicate a legible answer for each question.
- You are allowed any reasonable number of textbooks and quantity of notes as references. Otherwise, no notes, aides, or electronic equipment are allowed.
- Good luck!

UNIVERSITY REGULATIONS

1. Each candidate must be prepared to produce, upon request, a UBCcard for identification.
2. Candidates are not permitted to ask questions of the invigilators, except in cases of supposed errors or ambiguities in examination questions.
3. No candidate shall be permitted to enter the examination room after the expiration of one-half hour from the scheduled starting time, or to leave during the first half hour of the examination.
4. Candidates suspected of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action:
   - having at the place of writing any books, papers or memoranda, calculators, computers, sound or image players/recorders/transmitters (including telephones), or other memory aid devices, other than those authorized by the examiners;
   - speaking or communicating with other candidates; and
   - purposely exposing written papers to the view of other candidates or imaging devices. The plea of accident or forgetfulness shall not be received.
5. 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.
6. Candidates must follow any additional examination rules or directions communicated by the instructor or invigilator.

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Total
6	4	8	10	12	10	10	60

This page intentionally left blank.

If you put solutions here or anywhere other than the blank provided for each solution, you must *clearly* indicate which problem the solution goes with and also indicate where the solution is at the designated area for that problem's solution.

1 Representation Schemes [6 marks]

We wish to represent propositional logic statements in binary. Our representation uses 4-bit values to represent four symbols—$\sim$, $\wedge$, $\vee$, (, and)—and numbered variables like p_0, p_1, p_2, and so forth. Symbols are represented according to the following table:

Symbol	Representation
$\sim$	1000
$\wedge$	1010
$\vee$	1011
(	1100
)	1101

Variables are represented by 4-bit patterns that start with a 0, where a particular variable p_i is represented by the unsigned binary representation of the number i. (So, for example, $p_3 = 0011$.)

For example, the logical expression $\sim(p_0 \wedge p_1)$ would translate to 100011000000101000011101 as follows:

$\sim$	(	p_0	$\wedge$	p_1	)
1000	1100	0000	1010	0001	1101

1. Circle the best reason to reserve bit patterns starting with 1 only for symbols. **[2 marks]**

   (a) The representation scheme couldn't have been done any other way.

   (b) This scheme makes it easy to tell symbols from variables.

   (c) This scheme maximizes the number of variable numbers represented, given the length of the patterns and the number of symbols represented.

   (d) All of these are good reasons.

   Solution: This scheme makes it easy to tell symbols from variables (by looking at the first bit). Other schemes could be imagined including many that do not make it easy to distinguish symbols from variables; for example, unused patterns starting with 1 could encode additional variable numbers (e.g., 1111 might encode p_8)!

2. How many different variables can be represented in this scheme? **[1 mark]**

   Solution: The last three bits of a pattern starting with 1 encode the variable number, allowing 8 variables numbered 0 through 7.

3. We represent variable numbers as unsigned numbers. Circle the best reason *not* to have the variable numbers be signed? (That is, the last three bits of any 4-bit pattern that starts with a 0 would be the 3-bit signed binary representation of the variable's number.) **[1 mark]**

   (a) The four-bit patterns for variables start with 0; so, we can't use signed numbers.

   (b) Signed numbers are generally more complex to work with than unsigned.

   (c) We would not be able to represent as many distinct variables using signed numbers.

   (d) All of these are good reasons.

Solution : Signed numbers are generally more complex to work with than unsigned; so, why bother switching when we'd still have the same total number of variables (8 numbered -4 through 3)? We *could* use signed, however. We'd just stipulate that the rightmost three bits were a signed number so that, e.g., 0110 would be p_{-2}.

4. The hexadecimal values 6 and C each represent a single variable or symbol in our encoding scheme. What are they? **[2 marks]**

Solution : 6 is 0110 or p_6. C is 1100 or (.

2 Exploring Consequences of Propositional Logic Statements [4 marks]

Consider the following premises:

$q \wedge u$
$\sim(r \vee \sim p)$
$q \vee s$
$s \rightarrow (r \wedge p)$
$u \leftrightarrow (x \vee w)$
$x \rightarrow w$

For each of the following expressions, indicate whether—given the premises above—we know that the expression is true, know that the expression is false, know the expression is both true and false, or don't know the value of the expression. **[1 mark per question]**

1. u

Value: TRUE FALSE BOTH UNKNOWN

Solution : u is true, as we can see from the first premise (by simplification).

2. x

Value: TRUE FALSE BOTH UNKNOWN

Solution : x's value is unknown. Consider that $x = T, w = T, u = T$ and $x = F, w = T, u = T$ are both consistent with the premises above. So, we don't know whether x is true or false.

3. $x \vee p$

Value: TRUE FALSE BOTH UNKNOWN

Solution : $x \vee p$ is true. We know p is true from the second premise (after applying De Morgan's Law). If p is true then $x \vee p$ must be true.

4. $x \wedge \sim w$

Value: TRUE FALSE BOTH UNKNOWN

Solution : $x \wedge \sim w$ is false. It's the negation of the last premise $x \rightarrow w$ in disguise. $(x \wedge \sim w \equiv \sim(\sim x \vee w) \equiv \sim(x \rightarrow w)$.)

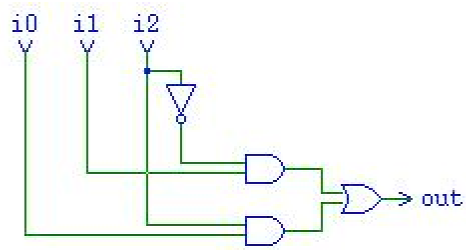
3 Circuit Design [8 marks]

Sketch a circuit that takes a 3-bit unsigned binary number as input and outputs true if and only if the input is a prime (2, 3, 5, or 7). Use only inverters and two-input AND, OR, NAND, NOR, XOR, and XNOR gates in your solution. (Note: Show your work to receive partial credit. . . and help you avoid mistakes!)

Solution : There are many possible solutions. We let the input be the three bit number $i_2 i_1 i_0$ and then use a truth table to look for a pattern to implement:

i	i_2	i_1	i_0	out
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Now, note that when i_2 is false, out is true exactly when i_1 is true. Wen i_2 is true, out is true exactly when i_0 is true. So, we build a circuit for $(\sim i_2 \wedge i_1) \vee (i_2 \wedge i_0)$:



4 Circuits, Correctness, and Equivalency [10 marks]

This problem focuses on a circuit to multiply two 2-bit unsigned numbers. The result is a 3-bit unsigned number plus an overflow flag. If the product of the inputs fits in 3 bits, the 3-bit output is the product and the overflow flag is false. Otherwise, the 3-bit output's value doesn't matter and the overflow flag is true.

The following truth table correctly specifies the circuit's behaviour, *but is not necessarily the only correct specification!* In the truth table, we multiply $x * y$ to produce z. For clarity, the truth table includes columns for x, y, and z's values as decimal numbers as well as binary numbers.

x	x_1	x_0	y	y_1	y_0	z	z_2	z_1	z_0	$overflow$
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0
0	0	0	2	1	0	0	0	0	0	0
0	0	0	3	1	1	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	1	1	0	0	1	0
1	0	1	2	1	0	2	0	1	0	0
1	0	1	3	1	1	3	0	1	1	0
2	1	0	0	0	0	0	0	0	0	0
2	1	0	1	0	1	2	0	1	0	0
2	1	0	2	1	0	4	1	0	0	0
2	1	0	3	1	1	6	1	1	0	0
3	1	1	0	0	0	0	0	0	0	0
3	1	1	1	0	1	3	0	1	1	0
3	1	1	2	1	0	6	1	1	0	0
3	1	1	3	1	1	9*	0	0	0	1

(The last entry in z's column is 9 because $3 * 3 = 9$; however, the last $z_2z_1z_0$ entry is not equal to 9.)

Answer the following questions about this problem:

1. Which of the following propositional logic statements correctly describes the output z_2? **Circle all correct answers. [4 marks]**

   (a) $(x_1 \wedge \sim x_0 \wedge y_1 \wedge \sim y_0) \vee (x_1 \wedge \sim x_0 \wedge y_1 \wedge y_0) \vee (x_1 \wedge x_0 \wedge y_1 \wedge \sim y_0)$

   (b) $(x_1 \oplus x_0) \wedge y_1$

   (c) $x_1 \wedge y_1$

   (d) $\sim(x_1 \wedge x_0 \wedge y_1 \wedge y_0)$

   Solution: (a) is the Disjunctive Normal Form implementation of the circuit as modeled by the truth table above. (b) is *incorrect*. Consider, for example, $x_1 = 1, x_0 = 1, y_1 = 1, y_0 = 0$. Then, $(1 \oplus 1) \wedge 1 = 0 \wedge 1 = 0$, but the output should be 1 in that row. (c) is *correct*. It does not match the truth table in the last row, but the output in the last row doesn't matter (as stated in the problem); so, that mismatch is irrelevant. (d) is incorrect. It will put a 1 in every row except the last, which obviously does not match the table.

2. Can two correct implementations of the overflow output be logically equivalent? Circle the best answer. **[2 marks]**

(a) Yes, they must be.

(b) They can be but aren't necessarily.

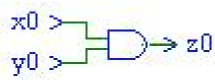
(c) No, they cannot be.

(d) There's not enough information to tell.

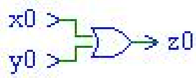
Solution : Yes, they must be logically equivalent. Every value of the overflow output is specified by the problem statement. So, every correct implementation must generate *exactly* the same values for the overflow output. Thus, every correct implementation is logically equivalent.

3. Which of the following circuits correctly implements the output z_0? **Circle all correct answers. [4 marks]**

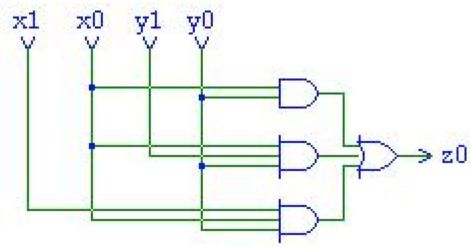
(a)



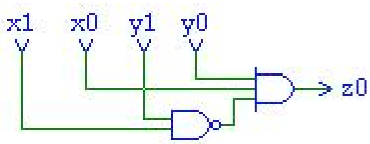
(b)



(c)



(d)



Solution : (a) is correct. As in the first part of this problem, the difference in the last row with the table's value for z_0 is irrelevant. (b) is incorrect. For example, it will produce a 1 in the table's second row. (c) is correct. The propositional logic model for that circuit is $(x_0 \wedge y_0) \vee (x_0 \wedge y_0 \wedge y_1) \vee (x_0 \wedge y_0 \wedge x_1)$. By applying absorption twice, we can see this is equivalent to $x_0 \wedge y_0$, which we saw above was correct. (d) is also correct. It simply eliminates the 1 from the last row of $x_0 \wedge y_0$'s truth table.

5 Propositional Logic Proof [12 marks]

1. **Without proceeding with the proof**, propose and justify two promising approaches to proving $p \wedge w$ (i.e., two alternative starting points in planning your proof) given the following premises. **[4 marks]**

   1. $r \vee s$
   2. $\sim s \rightarrow \sim x$
   3. $(\sim y \wedge r) \rightarrow p$
   4. $\sim s$
   5. $\sim y \vee s$
   6. $w \vee x$
   7. $y \rightarrow q$

   Solution : There are *many* legitimate approaches! We provide two for illustration.

   Approach #1:

   Solution : To prove $p \wedge w$, we must prove w. The only statement that mentions w is $w \vee x$; so, we should try prove $\sim x$.

   Approach #2:

   Solution : q is mentioned in only one premise; so, it's unlikely in general to (and cannot in this case) contribute to our solution. So, ignore the last premise and focus on the others.
   Note that a common solution was to write a proof. We specifically asked you *not* to do this; so, even though a correct proof suggests you must have thought of approaches, we did not give full marks for such an answer.
   Another common but problematic answer was to "use inference and logical equivalences rules to derive the conclusion from the premises". That's true but not responsive to this particular question (i.e., ignores the context). So, unless it was paired with more information, we did not give it credit.

2. Complete the following formal propositional logic proof that $\sim p \wedge q$ follows from the premises listed below. Use only logical equivalence laws and rules of inference in Epp Chapter 1 plus our "definition of conditional" and "definition of biconditional". You need not explicitly show steps that rely only on the double negation, commutative, and associative logical equivalence rules. **[6 marks]**

1.	$\sim(u\vee \sim q \vee s)$	premise
2.	$\sim r \leftrightarrow (\sim s \vee p)$	premise
3.	$a \rightarrow (b \vee c)$	premise
4.	$p \rightarrow u$	premise
5.	$q \vee r$	premise
6.	$(\sim r \rightarrow (\sim s \vee p)) \wedge ((\sim s \vee p) \rightarrow \sim r)$	Def'n of bicond on 2

Solution : The starting point we provided turns out to lead to a longer than necessary proof. We'll present both the shorter and longer versions here.

Short version:

7.	$\sim u \wedge q \wedge \sim s$	De Morgan's on 1
8.	q	simplification on 7
9.	$\sim u$	simplification on 7
10.	$\sim p$	modus tollens on 9 and 4
11.	$\sim p \wedge q$	conjunction on 10 and 8

Long version:

7.	$(\sim s \vee p) \rightarrow \sim r$	simplification on 6
8.	$\sim u \wedge q \wedge \sim s$	De Morgan's on 1
9.	$\sim s$	simplification on 8
10.	$\sim s \vee p$	addition on 9
11.	$\sim r$	modus ponens on 10 and 7
12.	q	elimination on 5
13.	$\sim u$	simplification on 8
14.	$\sim p$	modus tollens on 13 and 4
15.	$\sim p \wedge q$	conjunction on 14 and 12

3. Consider the following possibly flawed proof that r follows from the two premises shown. Circle each line of the proof that *does not* follow from the indicated previous lines *or* indicate that no line is flawed. **[2 marks]**

1.	$\sim(p \wedge q)$	premise
2.	$(\sim p \wedge \sim q) \rightarrow r$	premise
3.	$\sim p$	specialization on 1
4.	$\sim p \wedge (\sim p \vee \sim q)$	absorption on 3
5.	$\sim p \vee \sim q$	specialization on 4
6.	r	modus ponens on 5 and 2

Solution : Line 3 does not follow from Line 1. You cannot in general apply a rule of inference to just part of a propositional logic statement. In this case, consider if $p = T, q = F$. Then, premise 1 is true but line 3 is false.

Line 4 *does* follow from line 3. Equivalence rules can be applied in either direction. (Try to assign values to p and q so the two statements have different truth values; it's not possible!)

Line 5 *does* follow from line 4. We could have gotten there more simply with addition from line 3, however.

Line 6 does not follow from lines 2 and 5 because line 5 doesn't match the antecedent (left-hand side) of line 2.

6 Grab Bag [10 marks]

1. Is the conclusion of a propositional logic proof **logically equivalent** to the conjunction of its premises? **[2 marks]**

   (a) Yes, it must be logically equivalent, and the proof formally establishes that it is.

   (b) Yes, it must be logically equivalent, but the proof does not establish that it is.

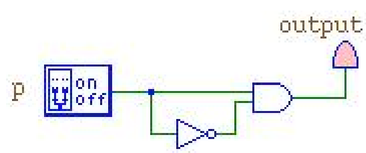
   (c) It may or may not be logically equivalent, and the proof does not establish whether it is.

   (d) No, it is not logically equivalent, but the proof does not formally establish that it's not.

   (e) No, it is not logically equivalent, and the proof formally establishes that it is not.

   Solution : It may or may not be logically equivalent, and the proof does not (in general) establish that it is. For example, if we conclude from the premises p and q that $p \wedge q$ is true, the conclusion is (trivially) logically equivalent to the premises ANDed together. If we conclude from the premise $p \wedge q$ that p is true, the conclusion is definitely *not* logically equivalent to the premise. In general, the conclusion of a propositional logic (or any non-equivalence) proof follows from but is not necessarily equivalent to its premises.

2. Consider the following circuit: **[3 marks]**



   According to our propositional logic model, the output of this circuit should always be false. Briefly explain what unmodeled behavior of real circuits makes it so the output will not always be false. (For full credit, indicate under exactly what circumstances this circuit's output becomes true.)

   Solution : Propositional logic does not model the time it takes for gates to calculate ("propagate") their results. In this case, if the input is false for a while, the output will also be false. When the input flips to true, both inputs to the AND gate will briefly be true (until the inverter "catches up"), and the AND gate will therefore briefly calculate and pass forward a true value.

3. Which of the following are equivalent to $p \leftrightarrow \sim q$? **Circle all correct answers. [3 marks]**

   (a) $(p \to q) \wedge (\sim p \to \sim q)$
   (b) $(p \to \sim q) \wedge (\sim q \to p)$
   (c) $p \oplus q$
   (d) $(p \wedge q) \vee (\sim p \wedge \sim q)$
   (e) $(\sim p \vee q) \wedge (p \vee \sim q)$
   (f) $(\sim p \wedge q) \vee (p \wedge \sim q)$

   Solution : It's worth trying to prove these equivalencies. Here's the list: $(p \to \sim q) \wedge (\sim q \to p)$, $p \oplus q$, and $(\sim p \wedge q) \vee (p \wedge \sim q)$.

4. Let D be the domain of 8-bit signed ints, *not* mathematical integers. (Bear in mind what you've learned about how arithmetic on signed binary numbers works, and where it differs from arithmetic on actual integers!)

   Is the following statement true? **[2 marks]**

   $\forall x \in D, \forall y \in D, (x > y) \to (x - y > 0)$.

   (a) Definitely true, which we can see by subtracting y from both sides of $x > y$.
   (b) Definitely true, as illustrated by $x = 1$ and $y = 127$.
   (c) Definitely true, but none of the reasons here prove it.
   (d) Definitely false, because if y is a large positive number, the subtraction may "overflow".
   (e) Definitely false, as illustrated by $x = 126$ and $y = -10$.
   (f) Definitely false, but none of the reasons here prove it.
   (g) Not enough information to tell.

   Powers of two reminder: In case they're handy for this question, here are several powers of two:

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}
1	2	4	8	16	32	64	128	256	512	1024

   Solution : Definitely false, and $x = 126, y = -10$ is a good counterexample. $126 > -10$; so, the left-hand side of the implication is true. However, the right-hand side is false because $126-(-10) = 126+10 > 127$, and 127 is the largest value representible with 8 bits. Therefore this addition "wraps around" to negative numbers. ($126 + 10 = -120$ for 8-bit signed ints.)

7 Predicate Logic [10 marks]

Consider the following predicates, defined as appropriate over the sets of all student IDs: S; the set of all courses: C; and the set of all programs of study P:

$$\begin{aligned}
HasCredit(s, c) &\equiv \text{The student with ID } s \text{ has earned passing credit for course } c. \\
Required(p, c) &\equiv \text{Students in program } p \text{ must earn passing credit for course } c \text{ to complete the program.} \\
HasRequirements(s, p) &\equiv \text{The student with ID } s \text{ has earned passing credit for all of program } p\text{'s required courses.}
\end{aligned}$$

1. Which of these correctly defines $HasRequirements$ in terms of $HasCredit$ and $Required$. **[2 marks]**

   (a) $HasRequirements(s, p) \equiv \exists c \in C, Required(p, c) \rightarrow HasCredit(s, c)$

   (b) $HasRequirements(s, p) \equiv \exists c \in C, Required(p, c) \wedge HasCredit(s, c)$

   (c) $HasRequirements(s, p) \equiv \forall c \in C, Required(p, c) \rightarrow HasCredit(s, c)$

   (d) $HasRequirements(s, p) \equiv \forall c \in C, Required(p, c) \wedge HasCredit(s, c)$

   (e) None of these

   Solution : $\forall c \in C, Required(p, c) \rightarrow HasCredit(s, c)$

2. According to the following definition, what does it mean for a course to be a "possible requirement"? (A literal translation is worth partial credit; for full credit succinctly explain the meaning in English.) **[2 marks]**

   $PossibleRequirement(p, c) \equiv \sim Required(p, c) \ \wedge \ \forall s \in S, HasRequirements(s, p) \rightarrow HasCredit(s, c)$

   Solution : A possible requirement is a course that is not required for a program but that all the students who have completed the requirements for the program have earned credit for. (A "universally popular" elective course with the students that complete this program.)

3. Imagine the set of all courses were CPSC 121, MATH 200, and BSKT 444. Alice has earned passing credit for CPSC 121 and MATH 200 but not BSKT 444. Bob has earned credit for CPSC 121 and BSKT 444 but not MATH 200. There are other students, but we don't know which courses they've taken. For each of the following statements, indicate whether it's known to be true, known to be false, or not known to be either true or false. Briefly justify your answer. **[2 marks per question]**

   (a) $\forall c \in C, \exists s \in S, HasCredit(s, c)$

   Value: TRUE FALSE UNKNOWN

   Justify:

   Solution : This is true, because Alice has taken CPSC 121 and MATH 200 and Bob has taken BSKT 444. So, for each course, some student has earned credit for it.

   (b) $\exists s \in S, \forall c \in C, HasCredit(s, c)$

   Value: TRUE FALSE UNKNOWN

   Justify:

Solution : This is unknown. Maybe some other student has taken all the courses, but Bob and Alice have not.

(c) $\forall s \in S, \forall c \in C, HasCredit(s, c)$

Value: TRUE FALSE UNKNOWN

Justify:

Solution : This is false. Both Bob and Alice are good counterexamples as neither of them has credit for every course.

This page intentionally left blank.

If you put solutions here or anywhere other than the blank provided for each solution, you must *clearly* indicate which problem the solution goes with and also indicate where the solution is at the designated area for that problem's solution.

This page intentionally left blank.

If you put solutions here or anywhere other than the blank provided for each solution, you must *clearly* indicate which problem the solution goes with and also indicate where the solution is at the designated area for that problem's solution.

This page intentionally left blank.
If you put solutions here or anywhere other than the blank provided for each solution, you must *clearly* indicate which problem the solution goes with and also indicate where the solution is at the designated area for that problem's solution.