CPSC 121 GROUP Midterm Exam #1
Wednesday, February 12, 2014, 17:30–18:35

[1] 1. Circle your personal choice: My 1% "discretionary mark" for CPSC 121 will be...

(a) ...based on tutorial attendance.

(b) ...added to the weight of pre-class quizzes.

(c) ...added to the weight of the final exam.

[4] 2. Circle the letter of the statement that best translates prompts (i)–(vi) using the definitions:

- $g \equiv$ I can say George's last name correctly.
- $s \equiv$ I can say Steve's last name without laughing.
- $h \equiv$ I love CPSC 121. (We chose h for "heart".)

(CPSC 121 instructor George's last name is hard to pronounce while Steve's is funny.)

i. I can't say George's last name correctly.

(a) $\sim g$
(b) g
(c) s
(d) $\sim s$

Solution : $\sim g$

ii. I love CPSC 121 but I cannot say Steve's last name without laughing.

(a) $h \oplus \sim s$
(b) $h \wedge \sim s$
(c) $h \vee \sim s$
(d) $h \rightarrow \sim s$

Solution : $h \wedge \sim s$
Saying "x but y" means x is true and y is true...but your listener might be surprised that y is true given that x is. That's interesting information in natural language, but we don't have a way to express it in predicate logic; so, this is just an "and".

iii. $\sim(s \wedge h)$

(a) I can't say Steve's last name without laughing, and I love CPSC 121.
(b) It's not true that I can say Steve's name without laughing and I love CPSC 121.
(c) I can say Steve's last name without laughing, or I love CPSC 121.
(d) I can't say Steve's last name without laughing, and I don't love CPSC 121.

Solution : "It's not true that I can say Steve's name without laughing and I love CPSC 121."
(If we used De Morgan's equivalence law on this, we'd get $\sim s \vee \sim h$, which is "I can't say Steve's last name without laughing, or I **don't** love CPSC 121.")

iv. $g \oplus s$

(a) I know how to pronounce George's last name, but I cannot say Steve's last name without laughing.

(b) I can say exactly one of the instructors' names without any problems.

(c) I know how to pronounce George's last name, or I can say Steve's last name without laughing, or both.

(d) I know how to pronounce George's last name but not Steve's.

Solution : Literally, this is "I know how to pronounce George's last name, or I can say Steve's last name without laughing, but not both."

None of these is a literal translation; which is the best translation?

Let's translate **back** from the answers. The first is $g \wedge \sim s$, not what we're looking for. The second is tricky; come back to it. The third is $g \vee s$, using **inclusive** or. The last is either not translatable (because we haven't said anything about pronouncing Steve's name) or we might loosely say it's $g \wedge \sim s$, the same as the first!

What about the second one? The "problems" we have are prononciation and laughter. So, this says we can say one instructors name or the other, but not both. That's a good match with our target logical statement.

[3] 3. Each statement below is a "contingency": true for some assignments of truth values to the variables and false for others. For each statement, give an assignment of truth values that makes the statement true and another that makes it false.

Solution :

	Statement	**True** assignment	**False** assignment
1.	$\sim a$	$a = F$	$a = T$
2.	$\sim(p \vee \sim q)$	$p = F, q = T$	$p = T, q = T$
3.	$p \leftrightarrow (q \wedge (\sim p \vee \sim r))$	$p = F, q = F, r = F$	$p = T, q = T, r = T$

Other solutions to 2/false: anything *other than* the 2/true solution.

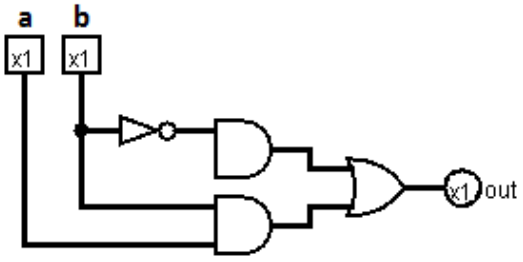
All solutions to 3/true: anything where the left and right sides have the same truth value; so, $p = F, q = F$ and any value of r, or $p = T, q = T, r = F$.

All solutions to 3/false: anything besides the 3/true solutions. So, $p = T, q = T, r = T$ and any other assignment that has different values for p and q.

[3] 4. Draw a direct translation of the following propositional logic expression into a circuit; label the output `out`.

$$(a \wedge \sim b) \vee (a \wedge b)$$

Solution :



[3] 5. Consider the **rule of inference**:

$$\begin{array}{l} p \to q \\ \underline{\sim q} \\ \therefore \sim p \end{array}$$

For each of the following, circle **APPLIES** to indicate the rule **can** be applied **directly** to the given statements or **CANNOT APPLY** to indicate it cannot. If the rule can be applied, provide the resulting statement.

NOTE: each problem starts with **two** statements to which we may be able to apply the rule.

Solution :

1. $\sim(a \wedge b) \to c, \sim c$ **APPLIES** with result: $\sim\sim(a \wedge b) \equiv a \wedge b$
2. $a \to (b \vee c), \sim c$ **CANNOT APPLY** (because $\sim c \not\equiv \sim(b \vee c)$)
3. $g \vee (h \to i), \sim i$ **CANNOT APPLY** (b/c cannot apply the rule *part* of the stmt)

[6] 6. Complete the following table where each row shows a single integer value represented as a "normal" decimal number, a 6-bit unsigned binary number, a 6-bit signed binary number, and a 2-digit unsigned hexadecimal number.

Fill in each blank, unshaded entry. **Write N/A if a value cannot be represented** in some column.

Solution :

Decimal	6-bit unsigned	6-bit signed	2-digit unsigned hexadecimal
11	**001011**	**001011**	**0B**
-1	**N/A**	**111111**	
-6		111010	
7		000111	
37	**100101**		25
	100000	**N/A**	**20**

[11] 7. Consider the following definitions describing a world of time travelers who may observe events in different orders:

- $I \equiv \{\mathrm{e}_1, \mathrm{e}_2, \mathrm{e}_3, \ldots\}$ is the set of "events" (important things that have happened)
- $P \equiv \{\mathrm{Adric}, \mathrm{Ben}, \mathrm{Clara}, \mathrm{Doc}, \ldots\}$ is the set of people
- $\mathrm{ObsOrder}(p, i, j)$ means person p observes that event i occurs before event j

- $\mathrm{Before}(i, j)$ means event i actually happened before event j
- $\mathrm{At}(p, i)$ means person p was at event i
- $\mathrm{Met}(p, q)$ means person p met person q

State the following in predicate logic:

(a) Doc has met himself.

Solution : $\mathrm{Met}(\mathrm{Doc}, \mathrm{Doc})$

(b) No event actually happened before itself.

Solution : $\forall i \in I, \sim \mathrm{Before}(i, i) \equiv \sim \exists i \in I, \mathrm{Before}(i, i)$

Assuming—for just this part—that the statements above are true, evaluate the truth of the following statements. (Circle one of **TRUE**, **FALSE**, or **UNKNOWN**.)

(a) **TRUE** **FALSE** **UNKNOWN** $\mathrm{Before}(\mathrm{e}_1, \mathrm{e}_1)$

Solution : **FALSE** (because we know $\forall i \in I, \sim \mathrm{Before}(i, i)$)

(b) **TRUE** **FALSE** **UNKNOWN** $\mathrm{Met}(\mathrm{Doc}, \mathrm{Clara})$

Solution : **UNKNOWN**

(c) **TRUE** **FALSE** **UNKNOWN** $\mathrm{At}(\mathrm{Doc}, \mathrm{e}_1) \rightarrow \mathrm{Met}(\mathrm{Doc}, \mathrm{Doc})$

Solution : **TRUE** (because we know $\mathrm{Met}(\mathrm{Doc}, \mathrm{Doc})$, which makes the whole conditional true)

Now, define the following predicates using $\mathrm{ObsOrder}$, Before, At, and Met:

(a) $\mathrm{Streamed}(i, j)$ means that event i and j actually happened in some order: one or the other first (but not both or neither).

Solution : $\mathrm{Streamed}(i, j) \equiv \mathrm{Before}(i, j) \oplus \mathrm{Before}(j, i)$

Note that $\oplus$ and not $\vee$ is necessary to avoid the "both" part (though there are other ways to express $\oplus$ of course!). Also note that neither i nor j should be quantified!

Also note that it's not necessary to state $i \neq j$ here. First, the definition doesn't explicitly require it; so, we should only add it with good reason! Second, it turns out that if $i = j$, we can derive a contradiction anyway. Can you see why?

(b) $\mathrm{Next}(i, j)$ means that event i actually happened before event j and no event actually happened between events i and j.

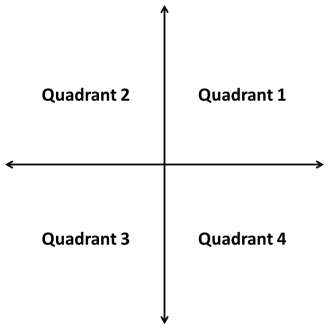
Solution : $\text{Next}(i,j) \equiv \text{Before}(i,j) \wedge \sim\exists k \in I, \text{Before}(i,k) \wedge \text{Before}(k,j)$

As usual, there are several (well, infinite) other ways to express the same idea, but one that does **not** work and is worth mentioning would be to move the $\exists$ quantifier to the left side of the statement. That would move it outside the $\sim$, which would very definitely change its meaning. (However, moving it out *and* changing it to a $\forall$ does work.)

[11] 8. A Cartesian coordinate is an x-value and a y-value like $(3, -2)$, which is 3 units in the positive direction (**right**, $\rightarrow$) on the x-axis and 2 units in the negative direction (**down**, $\downarrow$[1]) on the y-axis.

In the 16-bit "**VariableSplit**" representation, the first 3 bits are an unsigned binary number $a = a_1a_2a_3$ representing "the number of extra bits in the x-value". The 13 remaining bits $n_1n_2n_3 \ldots n_{13}$ are divided into two signed binary integers with lengths dependent on a. First the x-value, which is $3 + a$ bits long: $n_1n_2n_3 \ldots n_{3+a}$, and then the y-value, which is $10 - a$ bits long: $n_{4+a}n_{5+a} \ldots n_{13}$.

We name the "quadrants" of the xy-plane with the following chart, where anything on the x-axis or y-axis counts as "quadrant 0":



(a) What is the largest (positive) x-value representable with **VariableSplit**? (Write your answer in base 10.)

Solution : The x-value is a signed integer and has $3+a$ bits. a is an unsigned integer with 3 bits; so, it gets as large as 7. $3 + 7 = 10$; so, the x-value can be as large as 10 bits. The largest signed binary number representable in 10 bits is $2^{10-1} - 1 = 2^9 - 1$. $2^9 = 512$. Therefore..

511

(b) Give two distinct **VariableSplit** bit patterns that both represent the origin $(0, 0)$.

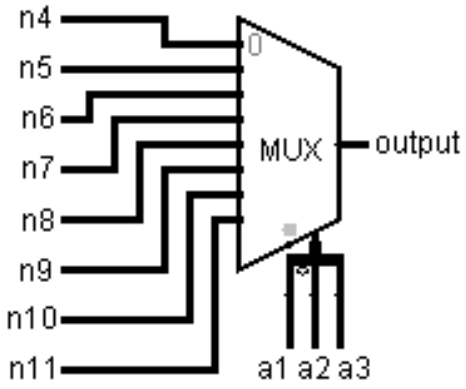
Solution : There are many possible answers, but the key idea is that the x- and y-values must both be zero (so all 13 bits of n are 0). a can be anything. Here's two possibilities:

[1] We incorrectly had "**up**, $\uparrow$" on the exam and gave credit for answers to (d) that considered either case.

0000000000000000
1110000000000000

(c) **Using a 1-of-8 multiplexer** sketch a circuit that takes as input bits n_4, n_5, n_6, n_7, n_8, n_9, n_{10}, n_{11}, and a_1, a_2, and a_3 you need from a **VariableSplit** value and produces the value of the first bit in the y-value (labeled n_{4+a} above). *Hint: a correct circuit is not very complex!*

Solution : The key insight is that we just need to "pick out" or "select" the correct bit from among $n_4 \ldots n_{11}$. The correct bit is the a^{th} of this group. So, we use a as the control or select bits to the multiplexer and the 8 given n bits as the inputs, and the output is what we want:



(d) **Design** a circuit (i.e., give a propositional logic formula for each bit of the output $q = q_1q_2q_3$ but do not draw the circuit diagram) that takes as input the first bit of the x-value x_1 and the first bit of the y-value y_1 from a **VariableSplit** value and produces the quadrant that coordinate is in, **assuming it is not in quadrant 0**. Your circuit's output should be an unsigned binary number 1–4.

Note: your circuit gets x_1 and y_1 as input; it doesn't need to figure out what they are!

Solution : Because the point cannot be in quadrant 0, the first bit tells us whether the value is positive or negative. We use an augmented truth table to proceed:

x_1	y_1	$x > 0$?	$y > 0$?	q	q_1	q_2	q_3
0	0	yes	yes	1	0	0	1
0	1	yes	no	4	1	0	0
1	0	no	yes	2	0	1	0
1	1	no	no	3	0	1	1

As always, we solve the three bits of output as three separate problems.

q_1 has only one row that's true: $q_1 = \sim x_1 \wedge y_1$.

q_2 has two rows that are true, both where x_1 is true: $q_1 = x_1$ (or, more verbosely, $(x_1 \wedge \sim y_1) \vee (x_1 \wedge y_1)$).

q_3 has two rows that are true; these happen to have the truth table $x_1 \leftrightarrow y_1 \equiv \sim(x_1 \oplus y_1)$, but we'll give the statement we can read off the truth table with our algorithmic approach: $q_3 = (\sim x_1 \wedge \sim y_1) \vee (x_1 \wedge y_1)$.

[7] 9. Complete the following propositional logic proof to prove u:

1. $q \wedge m$	premise
2. $\sim r \vee \sim(m \wedge \sim u)$	premise
3. $q \rightarrow (r \wedge s)$	premise

Solution : As always, in our scratch work, we work backward from the solution, play with the premises, and so forth. Here's our final solution:

4. q	by SPEC on 1
5. $r \wedge s$	by MP on 3 and 4
6. r	by SPEC on 5
7. m	by SPEC on 1
8. $\sim r \vee \sim m \vee u$	by DM on 2
9. $\sim m \vee u$	by ELIM on 6 and 8
10. u	by ELIM on 7 and 9

[6] 10. Consider the following propositional logic statement:

$((p \wedge q) \rightarrow (\sim q \wedge r)) \wedge p$

Simplify this statement, and **prove with a logical equivalence proof** that the original statement is equivalent to your simplified form.

Hint: the statement's simplest form has only two variables, each occuring only once.

PROOF:

Solution :

$((p \wedge q) \rightarrow (\sim q \wedge r)) \wedge p$	$\equiv (\sim(p \wedge q) \vee (\sim q \wedge r)) \wedge p$	by IMP
	$\equiv (\sim p \vee \sim q \vee (\sim q \wedge r)) \wedge p$	by DM
	$\equiv (\sim p \vee \sim q) \wedge p$	by ABS
	$\equiv (\sim p \wedge p) \vee (\sim q \wedge p)$	by DIST
	$\equiv F \vee (\sim q \wedge p)$	by NEG
	$\equiv \sim q \wedge p$	by I

[5] 11. **Two extra-fun BONUS questions just for groups!** (These are worth bonus marks... but finish the rest of the exam first; it's worth **WAY** more points-per-unit-effort than the bonus.)

Refer to the definitions in the problem about time-travelers and write the statement "everyone who was at the beginning of time (the event that actually happened first) met each other". For bonus credit the statement must be written both correctly and concisely and must not suggest that anyone met themselves.

Solution : This is much easier with a helper. $\text{First}(i)$ means event i actually happened before any other event.

$\text{First}(i) \equiv \forall j \in I, i \neq j \rightarrow \text{Before}(i, j)$

Now:

$\forall i \in I, \forall p_1 \in P, \forall p_2 \in P, (\text{First}(i) \wedge At(p_1, i) \wedge At(p_2, i) \wedge p_1 \neq p_2) \rightarrow Met(p_1, p_2)$

Refer to the documentation on our 16-bit **VariableSplit** Cartesian coordinate representation and give the coordinates of the point furthest from the origin that is representable as a **VariableSplit** number. Briefly justify your answer.

Solution : Distance from the origin is $\sqrt{x^2 + y^2}$. For a fixed split, we'd want the most distant x value and the most distant y-value, combined, which would certainly be in quadrant 3 (since we can represent one "extra" negative number in a signed binary representation).

However, we also need to decide what value of a yields the most distant point. Consider that putting **ALL** 16 bits into the x coordinate would yield a point (in the negative x direction) 2^{15} from the origin, **MUCH** further than $\sqrt{(2^7)^2 + (2^7)^2} = \sqrt{2^{14} + 2^{14}} = \sqrt{2^{15}} = 2^7 \cdot \sqrt{2}$. We want to put as many bits as possible into one coordinate. (Another way to think of this is that the range of the coordinate grows exponentially with the number of bits added, which certainly dominates the factor of no more than $\sqrt{2}$ we lose by minimizing the other coordinate's range.)

Which coordinate can get the most bits? x can get as many as $3 + 7 = 10$ bits; y can get as many as $10 - 0 = 10$ bits; so, it doesn't matter which we choose. (There's actually two solutions.) We'll put our money on x. Thus, we need $a = 7$, 10 x bits, 3 y bits, and the most negative values for each length. That would be $x = -2^9, y = -2^2$ or the coordinate:

$(-512, -4)$