CPSC 121 2016W2 Midterm 2 Sample Solutions

March 20, 2017

DRAFT sample solution. We may make some edits after marking!

1 Pred Logic to Proof Structure [6 marks]

Consider the following theorem:

$$(\forall x \in A, P(x,x)) \to (\exists a \in A, \forall b \in A, P(a,b) \to \exists c \in A, b \neq c \land P(b,c))$$

There is **not enough information** about A and P to prove this theorem. Instead, lay out as much of the structure of a direct proof of this theorem as you can. **Whenever you choose a specific value for a variable, specify what (if anything) this choice can depend on.**

Please list each distinct step as its own bullet point so it's easy for us to follow your proof structure! We've started you with the first bullet point and completed step and the second bullet point.

Proof:

- Assume $\forall x \in A, P(x,x)$.
-

NOTE: you may need **much** less than the full page on this problem.

1.1 Sample Solution

Proof:

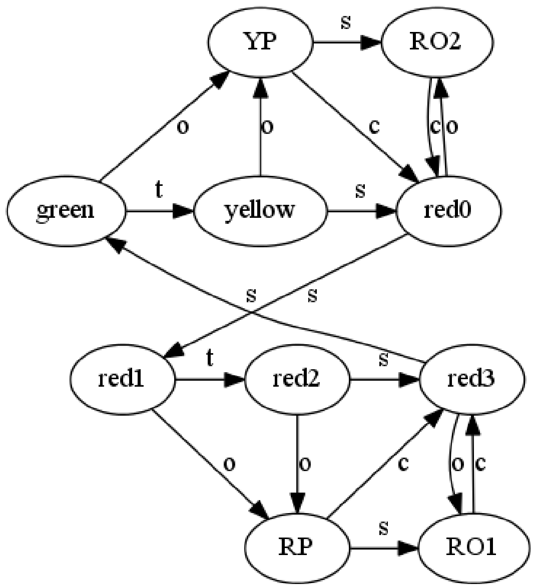
- Assume $\forall x \in A, P(x,x)$. (Note that this means we can select any element of A we like and know that P is true when we plug that element into both parameters of P.)
- Choose $a =$ _______. Note that a must be an element of A and is otherwise up to us! (It can "depend" on x, although that's not really meaningful, since x can be any value at all!)
- Without loss of generality, let b be an arbitrary element of A.
- Assume $P(a,b)$ is true.
- Choose $c =$ _______. Note that c must be an element of A, and its value can depend on b. We should choose c so that it is not equal to b. (Less importantly, as above c can depend on x, and c can also depend on a.)
- Show that $P(b,c)$ is true.

2 DFA to Circuit Layout [11 marks]

We now expand our traffic light DFA—as shown below—to handle an "ambulance override" that forces all directions to show red lights while an ambulance uses the intersection.

There are four new states: `RP` (red preparing for override, with the light red) and `RO1` (red override 1, with the light red) and `YP` (yellow preparing for override, with the light yellow) and `RO2` (red override 2, with the light red). There are two new inputs: override (`o`) starts the override and clear (`c`) ends it.

WARNING: To keep the diagram clean, we have omitted all "self-loops" from this DFA. That means there is no arrow in the DFA for any time an input letter would lead the DFA to transition **back to the same state**.



1. Imagine this DFA starts in the state `green` and processes the following input stream `tstssocs`. Indicate what state the DFA is in at the end of this input stream. **[2 marks]**

   REMEMBER that where an arrow is missing for an input at a particular state, that means the input causes the DFA to transition back to that state.

2. Briefly justify why this DFA will require four D Flip-Flops to translate it into a circuit. **[2 marks]**

3. Fill in the blanks in the following partial design of a circuit corresponding to this DFA. You are working **only** on truth table rows where the current state is `green` or `yellow` and the part of the circuit that updates the leftmost bit of the next state based on those two cases. We use the following representations for states and inputs:

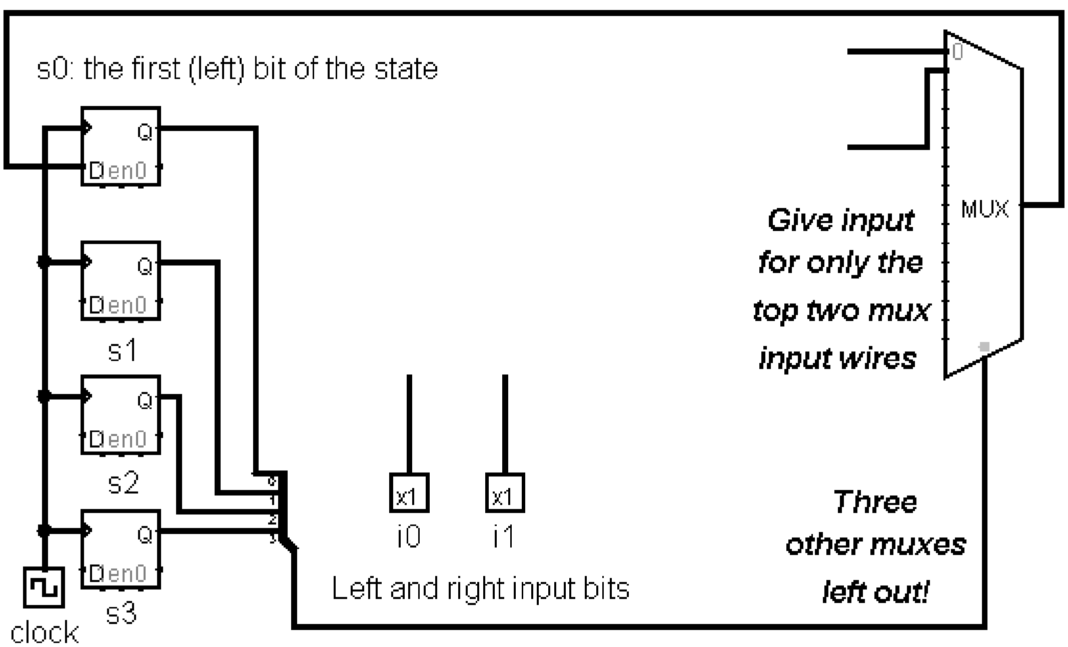
State:	`green`	`yellow`	`red0`	`red1`	`red2`	`red3`	`RP`	`RO1`	`YP`	`RO2`
Representation:	0	1	2	3	4	5	6	7	8	9

Input	`s`	`c`	`o`	`t`
Representation	0	1	2	3

Here is a partial truth table for this circuit. *Suggestion:* fill in the next state **names** first. **[4 marks]**

State	Input	State (as s0 s1 s2 s3)	Input (as i0 i1)	Next state (in binary)	Next state (name)
`green`	`s`	0 0 0 0	0 0		
`green`	`c`	0 0 0 0	0 1		
`green`	`o`	0 0 0 0	1 0		
`green`	`t`	0 0 0 0	1 1		
`yellow`	`s`	0 0 0 1	0 0		
`yellow`	`c`	0 0 0 1	0 1		
`yellow`	`o`	0 0 0 1	1 0		
`yellow`	`t`	0 0 0 1	1 1		

Here is a partial circuit. You should follow the instructions in the circuit to provide input for only the top two wires of the one multiplexer shown: **[3 marks]**



2.1 Sample Solution

1. Starting in `green` and processing `tstssocs`, we transition through these states:
   - `green` (start)
   - `yellow`
   - `red0`
   - `red0` (self-loop, not pictured on the DFA)
   - `red1`
   - `red1` (self-loop again)
   - `RP`

- `red3`
- `green`

2. This DFA requires 4 DFF's to translate into a circuit because it has 10 states, which is too many to represent with 3 bits (i.e., $2^3 = 8$ values) but fine for 4 bits (i.e., $2^4 = 16$ values).

   We didn't ask this, but... this DFA requires 2 bits of input (i.e., two input wires) to translate to a circuit because it has 4 inputs, which is too many for one wire ($2^1 = 2$ values) but fine for two wires ($2^2 = 4$ values).

3. Below are completed versions of the partial truth table and partial circuit.

   Note that we use the following representations for states and inputs:

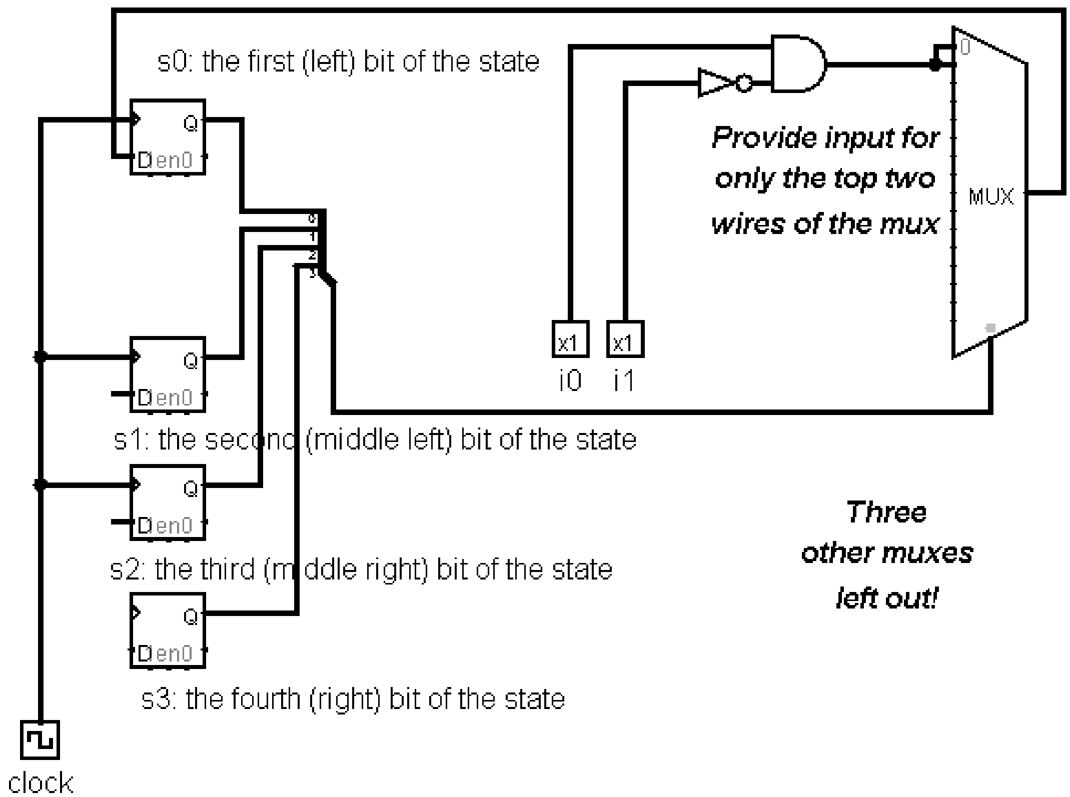
State:	`green`	`yellow`	`red0`	`red1`	`red2`	`red3`	`RP`	`RO1`	`YP`	`RO2`
Representation:	0	1	2	3	4	5	6	7	8	9

Input	`s`	`c`	`o`	`t`
Representation	0	1	2	3

Here is a (completed) partial truth table for this circuit:

State	Input	State (as s0 s1 s2 s3)	Input (as i0 i1)	Next state	Next state
`green`	`s`	0 0 0 0	0 0	0 0 0 0	`green`
`green`	`c`	0 0 0 0	0 1	0 0 0 0	`green`
`green`	`o`	0 0 0 0	1 0	1 0 0 0	`YP`
`green`	`t`	0 0 0 0	1 1	0 0 0 1	`yellow`
`yellow`	`s`	0 0 0 1	0 0	0 0 1 0	`red0`
`yellow`	`c`	0 0 0 1	0 1	0 0 0 1	`yellow`
`yellow`	`o`	0 0 0 1	1 0	1 0 0 0	`YP`
`yellow`	`t`	0 0 0 1	1 1	0 0 0 1	`yellow`

Here is a (completed) partial circuit:



3 Proof Structure to Pred Logic [7 marks]

The following is a portion of a proof remaining on a ~~blood-stained~~ peanut butter-stained sheet of paper after a hungry wolf attack. (Happily, the proof is double-spaced in case you want to take notes.)

> Without loss of generality, let s be an arbitrary snack. Assume that s is not peanut butter and that it masses at least 0.1 kilograms. We choose animal a to be... *the choice of A is unreadable.*
>
> Note that a likes s, as required. We then show... *the rest of the proof is unreadable*

Write down as much as possible of the original predicate logic theorem that is being proven.

If you need them, use the sets A of animals (such as `wolf`) and S of snacks (such as `PB` for peanut butter), and the predicate $L(x, y)$ meaning x likes y and the function $m(x)$ that gives the mass in kilograms of x (which can be an animal or snack).

NOTE: you may need **much** less than the full page on this problem.

3.1 Sample Solution

$$\forall s \in S, (s \neq \text{PB} \wedge m(s) \geq 0.1) \rightarrow \exists a \in A, L(a, s) \wedge \ldots$$

4 Direct Proof [7 marks]

Consider the following function: $f(n) = a^n - \frac{b^n}{3}$.

Prove the following theorem directly:

There is a negative real number b such that for all positive real numbers a and all natural numbers n, if $f(n)$ is an integer, then $f(n)$ is also the result of rounding a^n to the nearest integer.

Substantial partial credit is available for the structure of your proof. To finish the proof, here's a hint: you'll want $\frac{b^n}{3}$ to be less than $\frac{1}{2}$ and greater than $-\frac{1}{2}$. There's a **simple**, **integer** value for b that makes this work.

NOTE: you may need **much** less than the full page on this problem.

4.1 Sample Solution

Proof:

Choose $b = -1$.

WLOG, let a be an arbitrary positive real number and n be an arbitrary natural number.

Assume $f(n)$ is an integer.

$f(n) = a^n - \frac{b^n}{3}$. So, $a^n = f(n) + \frac{b^n}{3}$.

Since $b = -1$ and depending on whether n is even or odd, then $a^n = f(n) + \frac{1}{3}$ or $a^n = f(n) - \frac{1}{3}$. $\frac{1}{3} < 0.5$. So, $f(n) - 0.5 < a^n < f(n) + 0.5$. In other words, a^n is closer to the integer $f(n)$ than to either the integer below or above it. Thus, a^n rounds to $f(n)$. **QED**

This problem is loosely related to a formula for computing the n^{th} Fibonacci number. For the curious, the n^{th} Fibonacci number $F_n = (\frac{1+\sqrt{5}}{2})^n/\sqrt{5} - (\frac{1-\sqrt{5}}{2})^n/\sqrt{5}$. With $a = \frac{1+\sqrt{5}}{2}$ and $b = \frac{1-\sqrt{5}}{2}$, that's $\frac{a^n}{\sqrt{5}} - \frac{b^n}{\sqrt{5}}$. Because $|\frac{b^n}{\sqrt{5}}| < 0.5$ for all $n \geq 0$ and because all the Fibonacci numbers are integers, however, we can just round $\frac{a^n}{0.5}$ to the nearest integer to compute F_n rather than bothering with the $\frac{b^n}{\sqrt{5}}$ part.

5 Proof by Contradiction [5 marks]

Prove the following theorem **using a proof by contradiction**.

(**NO** credit is available for other styles of proof besides contradiction. Substantial partial credit is available for the structure of the proof. Don't overthink this problem! Your proof should be about 2–3 lines long.)

Theorem: For any pair of integers x and y, $5x + 10y \neq 1$.

NOTE: you may need **much** less than the full page on this problem.

5.1 Sample Solution

As instructed, we proceed by contradiction.

Proof:

Assume for contradiction that there are integers x and y such that $5x + 10y = 1$. Then, $5(x + 2y) = 1$ and $x + 2y = \frac{1}{5}$, but $x + 2y$ is an integer while $\frac{1}{5}$ is not. That's a contradiction.

QED

6 Induction [14 marks]

Prove by induction that $4^n + 5$ is divisible by 3 for all natural numbers $n \geq 0$. (That is, prove that $4^n + 5 = 3k$ or $4^n = 3k - 5$ for some integer k.)

(**NO** credit is available for other styles of proof besides induction. Substantial partial credit is available for the structure of the proof. Remember to clearly label the parts of your proof and to briefly justify any step you take that is not a basic arithmetic step. The marks in this problem are roughly divided into **4 marks** for the base case, **6 marks** for the structure of the inductive step/induction hypothesis, and **4 marks** for completing the proof.) **This page intentionally left (almost) blank. If you write answers here, you must CLEARLY indicate on this page what question they belong with AND on the problem's page that you have answers here.**

6.1 Sample Solution

We can focus on the recursive structure of natural numbers or of 4^n to structure our induction proof. Either way, we get a base case of $n = 0$ and a recursive case for $n > 0$ in which n depends on $n - 1$.

Proof:

BC: When $n = 0$, $4^n + 5 = 4^0 + 5 = 1 + 5 = 6 = 3 * 2$. So, $4^n + 5$ is divisible by 3.

IS: Consider an aritrary $n > 0$.

IH: Assume $4^{n-1} + 5 = 3k$ for some integer k.

Now, we must show $4^n + 5 = 3k'$ for some integer k'.

Note that $4^{n-1} = 3k - 5$.

Let $k' = 4k - 5$, which is an integer because k is (and the integers are closed under multiplication and subtraction).

Consider:

$$\begin{aligned}
4^n + 5 &= 4 * 4^{n-1} + 5 \\
&= 4 * (3k - 5) + 5 && \text{by the IH} \\
&= 12k - 20 + 5 \\
&= 12k - 15 \\
&= 3(4k - 5) \\
&= 3k' && \text{b/c } k' = 4k - 5
\end{aligned}$$

So, $4^n + 5$ is divisible by 3, which completes the inductive step and our induction proof. **QED**