Staff Solutions to Midterm Exam March 14

STAFF NOTE: TOPICS:
Normal Forms, Predicate Formulas Ch.3.4.1, 3.5-6 (omit 3.4.2)
Predicate Logic Ch.3.7
Sets \& Sequences Ch.4-4.2
Binary Relations Ch.4.3-5
Induction Ch.5-5.3
State Machines Ch.6-6.3
Stable matching Ch.6.4
Recursive Data, Structural Induction Ch.7-7.4
Recursive Games

Problem 1 (Injections) (9 points).
The definition of $A \text{ surj } B$ requires that there be a surjective function from $A$ to $B$. Suppose the function condition was dropped, so we have a simpler definition $A \text{ simpsurj } B$ iff there is a surjective $\geq 1$ in relation from $A$ to $B$. For each of the following items, give a simple description of the sets $B$ such that

(i) $\{2, 7\} \text{ simpsurj } B$.

(ii) $\{\emptyset\} \text{ simpsurj } B$.

(iii) $\emptyset \text{ simpsurj } B$.

COMMENTS:
- MQ_simpsurj
- part(b) of TP_simpinj
- From class 2/27/17

keywords = [ Functions Injections Surjections ]

Solution. If $A$ is nonempty, then $A \text{ simpsurj } B$ is true for all sets $B$, because there is an element $e$ in $A$, and the relation that has arrows from $e$ to every element of $B$ has $\geq 1$ in and is therefore surjective.

Also $\emptyset \text{ simpsurj } B$ is true iff $B = \emptyset$. 

2017, Albert R Meyer. This work is available under the terms of the Creative Commons Attribution-ShareAlike 3.0 license.
Problem 2 (Logical Formulas, Induction) (16 points).
Let’s say a predicate $P$ on the nonnegative integers is a plus2 predicate when the following assertion is true:

$$\forall n. \ P(n) \text{ IMPLIES } P(n + 2).$$

For each of the assertions (a)–(d) below, indicate whether the assertion is true for All plus2 predicates $P$, true for Some but not all plus2 predicates $P$, or Not true for any plus2 predicate $P$ by circling the correct letter below. For assertions which are true for Some plus2 predicates $P$ but not others, describe a plus2 predicate for which the assertion is true and another for which it is false. Do not include explanations.

(a) $P(1) \text{ IMPLIES } \forall n. \ P(n + 1)$

(b) $\lnot (P(0)) \text{ AND } \forall n \geq 1. \ P(n)$

(c) $\exists n. \exists m > n. \ [P(2n) \text{ AND } \lnot (P(2m))]$

(d) $\exists n. \exists m < n. \ [P(2n) \text{ AND } \lnot (P(2m))]$

Solution. A. This assertion says that if $P(1)$ holds, then $P(n)$ holds for all odd $n$. This is always true for a plus2 predicate.

Solution. S. Letting $P(n) := n \geq 1$ gives a plus2 predicate making the assertion true, and letting $P(n)$ be the always be true predicate gives a plus2 predicate that makes the assertion false since $\lnot (P(0))$ is false.

Solution. N. This assertion says that $P$ holds for some even number $2n$, but not for some other larger even number $2m$. However, if $P(2n)$ holds, we can invoke the plus2 property $n - m$ times to conclude $P(2m)$ also holds.

Solution. S. Letting $P(n) := n > 0$ is a plus2 predicate that makes this assertion true since $0 < 1$ and $P(2 \times 1) \text{ AND } \lnot (P(2 \times 0))$ is true. Letting $P(n)$ be the always be true predicate is a plus2 predicate that makes this assertion false because $\lnot (P(2m))$ is false for every $m$.

Problem 3 (Induction) (15 points).
Define

$$T_n := T_{n-1} + 2T_{n-2} \quad \text{(for } n \geq 2),$$

where $T_0 = T_1 = 1$.

Prove by induction that

$$T_n = \frac{2^{n+1} + (-1)^n}{3} \quad \text{(*)}$$

for all $n \geq 0$.

COMMENTS:
Solution. The induction hypotheses $P(n)$ will be equation (*).

**Base cases** ($n = 0, 1$)

$$T_0 = 1 = \frac{2^{(0+1)} + (-1)^0}{3}, \quad T_1 = 1 = \frac{2^{(1+1)} + (-1)^1}{3}.$$  
So $P(0)$ and $P(1)$ are true.

**Induction step.** For $n \geq 2$, we need to prove $P(n)$ assuming by strong induction that $P(k)$ is true for $0 \leq k < n$.

$$T_n := T_{n-1} + 2T_{n-2} \quad \text{(by def since $n \geq 2$)}$$

$$= \frac{2^{(n-1)+1} + (-1)^{n-1} + 2^{(n-2)+1} + (-1)^{n-2}}{3} \quad (P(n-1), P(n-2) \text{ are true by hypothesis})$$

$$= \frac{2^n + (-1)^{n-1} + 2^n + 2(-1)^{n-2}}{3}$$

$$= \frac{2(2^n) - (-1)^n + 2(-1)^n}{3}$$

$$= \frac{2^{n+1} + (-1)^n}{3}$$

This proves $P(n)$, completing the induction step.

It follows by strong induction that (*) holds for all $n \geq 0$, as desired.  

---

**Problem 4 (State Machines) (20 points)**.

The following problem is a twist on the Fifteen-Puzzle considered earlier.

Let $A$ be a sequence consisting of the numbers $1, \ldots, n$ in some order. A pair of integers in $A$ is called an out-of-order pair when the first element of the pair both comes earlier in the sequence, and is larger, than the second element of the pair. For example, the sequence $(1, 2, 4, 5, 3)$ has two out-of-order pairs: $(4, 3)$ and $(5, 3)$. We let $t(A)$ equal the number of out-of-order pairs in $A$. For example, $t((1, 2, 4, 5, 3)) = 2$.

The elements in $A$ can be rearranged using the Rotate-Triple operation, in which three consecutive elements of $A$ are rotated to move the smallest of them to be first.

For example, in the sequence $(2, 4, 1, 5, 3)$, the Rotate-Triple operation could rotate the consecutive numbers $4, 1, 5$, into $1, 5, 4$ so that

$$(2, 4, 1, 5, 3) \rightarrow (2, 1, 5, 4, 3).$$

The Rotate-Triple could also rotate the consecutive numbers $2, 4, 1$ into $1, 2, 4$ so that

$$(2, 4, 1, 5, 3) \rightarrow (1, 2, 4, 5, 3).$$

We can think of a sequence $A$ as a state of a state machine whose transitions correspond to possible applications of the Rotate-Triple operation.
(a) Argue that the derived variable $t$ is weakly decreasing.

**COMMENTS:**
- TP_sort_cyclic_shift_three
- related to CP_fifteen_puzzle
- ARM and CH, S14

**keywords** = ['state_machines', 'sorting', 'fifteen_puzzle', 'parity']

**Solution.** Suppose the *Rotate-Triple* operation is applied to three consecutive elements $a$, $b$, $c$ in $A$. This has no effect on the out-of-order pairs involving at most one of $a$, $b$, and $c$.

To analyze pairs where both elements are one of $a$, $b$, and $c$, there are two cases.

If $b$ is the smallest element, then $a$, $b$, $c$ get rearranged into $b$, $c$, $a$. This has the effect of reversing the two pairs $(a, b)$, $(a, c)$ into $(b, a)$, $(c, a)$. If $a < c$, this causes a net change of zero in $t(A)$, while if $c < a$, this causes a net decrease of two in $t(A)$.

If $c$ is the smallest, then $a$, $b$, $c$ get rearranged into $c$, $a$, $b$. This has the effect of reversing the two pairs $(a, c)$, $(b, c)$ into $(c, a)$, $(c, b)$, which similarly leaves $t(A)$ unchanged or decreased by two.

So in each case, $t$ is either constant or decreases, showing that $t$ is weakly decreasing.

(b) Prove that having an even number of out-of-order pairs is a preserved invariant of this machine.

**Solution.** This part follows directly from the argument in the previous part showing that $t$ changes by 0 or $-2$. So if the number of out-of-order pairs is even, then it stays even.

**Problem 5 (Stable Marriage) (15 points).**
A preserved invariant of the Mating Ritual is:

For every girl $G$ and every boy $B$, if $G$ is crossed off $B$’s list, then $G$ has a favorite suitor whom she prefers more than $B$.

Let Brad be some boy and Jen be any girl who is not his wife on the last day of the Mating Ritual. You may assume that everyone is married on this last day.

Use the invariant to explain why Brad and Jen are not a rogue couple. Conclude that the Mating Algorithm produces stable marriages.\(^1\)

**COMMENTS:**
- CP_mating_ritual_proof
- class or exam problem only: proof in book
- from: S07.cp5m, S08.cp5f(?)

**keywords** = ['stable_matching', 'Mating_ritual', 'invariant']

\(^1\)This proof was already given in the text. We think that someone with a basic understanding of the Mating Ritual will be able to reconstruct the proof for themselves. If you memorized that proof (we hope you didn’t; memorization is not a sensible approach to learning 6.042 class material) or already copied it onto your crib sheet, then you have lucked out and will nail this question.
Solution. Proof. To prove the claim, we consider two cases:

Case 1. Jen is not on Brad’s list. Then by the invariant, we know that Jen prefers her husband to Brad. So she’s not going to run off with Brad: the claim holds in this case.

Case 2. Otherwise, Jen is on Brad’s list. But since Brad works his way down his preference list until he finds a wife, his wife must be higher on his preference list than Jen. This means that Brad prefers his wife to Jen, so he’s also not going to run off with Jen: the claim also holds in this case.

So in any case, Brad will not be part of a rogue couple. Since Brad is an arbitrary boy, it follows that no boy is part of a rogue couple. Hence the marriages on the last day are stable.

Problem 6 (Propositional Formulas, Structural Induction) (25 points).
A class of propositional formulas called the Multivariable AND-OR (MVAO) formulas are defined recursively as follows:

Definition. Base cases: A single propositional variable, and the constants True and False are MVAO formulas.

Constructor cases: If $G, H \in$ MVAO, then $(G \text{ AND } H)$ and $(G \text{ OR } H)$ are MVAO’s.

For example,

$$(((P \text{ OR } Q) \text{ AND } P) \text{ OR } (R \text{ AND } \text{True})) \text{ OR } (Q \text{ OR } \text{False})$$

is a MVAO.

Definition. A propositional formula $G$ is False-decreasing when substituting the constant False for some occurrences of its variables makes the formula “more false.” More precisely, if $G^\f$ is the result of replacing some occurrences of variables in $G$ by False, then any truth assignment that makes $G$ false also makes $G^\f$ false.

STAFF NOTE: So $G$ is False-decreasing iff $[G^\f \text{ IMPLIES } G]$ is valid. However, this fact did not help me find a better proof than the one below. Let me know if you come up with one—ARM 3/9/17.

For example, the formula consisting of a single variable $P$ is False-decreasing since $P^\f$ is the formula False. The formula $G := \overline{P}$ is not False-decreasing since $G^\f$ is the formula False which is true even under a truth assignment where $G$ is false.

Prove by structural induction that every MVAO formula $F$ is False-decreasing.

COMMENTS:

- FP OR AND_recursive_multivar
- similar to CP_XOR_AND_recursive, CP_XOR_AND_formulas
- ARM 3/9/17

keywords = [ recursive structural_induction OR AND constant formula]

Solution. Proof. Base case:

- $(F$ is a single variable). This was just given above.

- $(F$ is a constant True or False). Then $F^\f$ is the same as $F$, so which trivially implies that $F^\f$ is false whenever $F$ is false.
**Constructor case:** \( (G \text{ is } [F \text{ AND } H]) \). Replacing some variable occurrences in \( G \) by \textbf{False} means that some occurrences in each of \( F \) and \( H \) are replaced by \textbf{False}. In other words, any \( G^f \) equals \( F^f \text{ AND } H^f \) for some \( F^f \) and \( H^f \).

Now by definition of \textbf{AND}, \( G^f \) is true under some true some assignment \( A \) iff both \( F^f \) and \( H^f \) are true under \( A \). By structural induction hypothesis, if \( F^f \) and \( H^f \) are true under \( A \), then \( F \) and \( H \) are true under \( A \). Hence \( F \text{ AND } H \), that is \( G \), is true under \( A \). This shows that \( G \) is \textbf{False}-decreasing.

**Constructor case:** \( (G \text{ is } [F \text{ OR } H]) \). As in the \textbf{AND} case, we have that \( G^f \) equals \( F^f \text{ OR } H^f \).

Now by definition of \textbf{OR}, \( G^f \) is true under some true some assignment \( A \) iff either \( F^f \) or \( H^f \) is true under \( A \). Now if \( F^f \) is true under \( A \), then by structural induction hypothesis, \( F \) is true under \( A \). Hence \( F \text{ OR } H \), that is \( G \), is true under \( A \). The same argument holds if \( H^f \) is true under \( A \). This shows that in any case, \( G \) is \textbf{False}-decreasing. \( \blacksquare \)