C241 Fall 08: Final Exam Review Solutions

The exam will be a mix of short-answer, true/false style problems and induction problems. It's not cumulative, it will focus on material covered since the second midterm and on the last two homeworks.

1) Use the language P defined below to answer the following questions.

	$P \subset V^+ V = \{a, b, c\}$
1.	$a \in P$
2a.	$u \in P \Rightarrow bub \in P$
2b.	$u \in P \Rightarrow ucu \in P$
3.	There is nothing else in P.

1a) Which of these strings are in the language P?

(i)	a, Ok (1)	(v)	aca, Ok (1,2b)
(ii)	bab, Ok (1,2a)	(vi)	aacaa, No ($aa \notin P$)
(iii)	b, No	(vii)	ababcbaba, No ($abab \notin P$)
(iv)	bcb, No ($b \notin P$)	(viii)	bacacacab, Ok (1[a],2b[aca],2b[acacaca],2a[bacacacab])

1b) Using structural induction proofs, prove that every element in P has the following properties:

Every $u \in P$ has an even number of b's.

Base Case: $u = a$

a has 0 b's, and 0 is an even number.

Induction Hypothesis: Say n is the number of b's in u . Assume n is even.

Induction Step 2a: Show bub has an even number of b's.

bub has $n + 2$ b's. Since n is even [IH], $n + 2$ is even as well.

Induction Step 2b: Show ucu has an even number of b's.

ucu has $2n$ b's (n b's from each copy of u). $2n$ is even for any n.

Every $u \in P$ has at least one a.

Base Case: $u = a$

a clearly has exactly one a.

Induction Hypothesis: Assume there is at least one a in u.

Induction Step 2a: Show bub has at least one a.

bub has the same number of a's as u. By our [IH] that's at least one.

Induction Step 2b: Show ucu has at least one a.

ucu has twice as many a's as u. Since our [IH] tells us that u has at least one a, ucu actually has at least two a's.

Every $u \in P$ has an odd number of letters.

Base Case: $u = a$

a has length 1, which is odd.

Induction Hypothesis: Say that u has n letters. Assume $n = 2q + 1$ for some integer q (in other words, assume n is odd) .

Induction Step 2a: Show bub has an odd number of letters.

bub has $n+2$ letters. By our [IH] $n+2 = (2q+1)+2$ which we can simplify to $2(q + 1) + 1$, which is by definition an odd number (since any number of the form $2m + 1$, where m is an integer, is odd).

Induction Step 2b: Show ucu has an odd number of letters.

ucu has $2n + 1$ letters. By our [IH] $2n + 1 = 2(2q + 1) + 1$, which is by definition an odd number, (since $(2q + 1)$ is an integer).

Every $u \in P$ is a palindrome (p and $reverse(p)$ are the same).

Base Case: $u = a$

Since a is a single letter, it's a palindrome by default .

Induction Hypothesis: Assume that u is a palindrome... in other words, assume that $u = reverse(u)$

Induction Step 2a: Show bub is a palindrome.

Since we're adding a b to both the left and right ends, when we reverse bub we get $reverse(bub) = b\, reverse(u)\, b$. So $bub = reverse(bub)$ if $u = reverse(u)$. Which is true by our [IH].

Induction Step 2b: Show ucu is a palindrome.

Since both the left and right sides of ucu are u, if $u = reverse(u)$ this will be a palindrome (the left and right side will be mirror images of each other). Fortunately, our [IH] tells us this is the case.

2) Use the language L, and the functions I, A, and R defined below to answer the following questions.

	$L \subset V^+ V = \{a, b, \dot{}\}$
1.	$\bullet \in L$
2a.	$u \in L \Rightarrow au \in L$
2b.	$u \in L \Rightarrow bu \in L$
3.	There is nothing else in L.

–

	$I : L \to L$
I1.	$I(\bullet) = \bullet$
I2a.	$I(au) = bI(u)$
I2b.	$I(bu) = aI(u)$

	$R : L \to L$
R1.	$R(\bullet) = \bullet$
R2a.	$R(au) = A(R(u), a\bullet)$
R2b.	$R(bu) = A(R(u), b\bullet)$

	$A : L \times L \to L$
A1.	$A(\bullet, v) = v$
A2a.	$A(au, v) = aA(u, v)$
A2b.	$A(bu, v) = bA(u, v)$

	$T : L \times L \to L$
T1.	$T(\bullet, v) = v$
T2a.	$T(au, v) = T(u, av)$
T2b.	$T(bu, v) = T(u, bv)$

2a) Label the following statements True or False:

i) $\bullet \in L$ True (1)
ii) $a \in L$ False
iii) $R(bb\bullet) = A(R(b\bullet), b\bullet)$ True
iv) $A(u, R(\bullet)) = \bullet$ False ($A(u, R(\bullet)) = A(u, \bullet) \neq \bullet$)
v) $A(v, A(R(u), a\bullet)) = A(v, A(u, a\bullet))$ False ($R(u) \neq u$)
vi) $I(ab\bullet) = bI(a\bullet) = baI(\bullet) = ba\bullet$ False (should be $I(ab\bullet) = bI(b\bullet) = baI(\bullet) = ba\bullet$)
vii) $I(u) = u \begin{bmatrix} a \\ b \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix}$ False (this would result in a string of all b's)

viii) $I(u) = u \begin{bmatrix} a, & b \\ b, & a \end{bmatrix}$ True (I replaces all the b's in u with a's , and all a's with b's)

2b) Write a recursive function (in the style of the functions I, A, and R above) which takes a string $u \in L$ and performs the substitution $u \begin{bmatrix} bb \\ b \end{bmatrix}$.

	$I : L \to L$
I1.	$I(\bullet) = \bullet$
I2a.	$I(au) = aI(u)$
I2b.	$I(bu) = bbI(u)$

2c) Prove that for all $u, v \in L$, $T(u, v) = A(R(u), v)$. In addition to the function definitions above, you can use the following three lemmas which were proved on the homework. (a 'lemma' is a small fact you prove in order to make a larger proof easier):

Lemma 1: For all $u, v, w \in L$, $A(A(u, v), w) = A(u, A(v, w))$

Lemma 2: For all $u \in L$, $A(u, \bullet) = u$

Lemma 3: For all $u, v \in L$, $R(A(u, v)) = A(R(v), R(u))$

Base Case: $u = \bullet$

Show: $T(\bullet, v) = A(R(\bullet), v)$

$T(\bullet, v) = A(\bullet, v)$ (R1)

$v = A(\bullet, v)$ (T1)

$v = v$ (T1)

Induction Hypothesis: Assume $T(u, v) = A(R(u), v)$ for all $v \in L$

(note: we add the 'for all $v \in L$' because we're only doing induction on u–we're leaving v as an open variable. And since we're not restricting the value of v, we're assuming our IH works regardless of what value comes after the comma in the T function.)

Induction Step 2a: Show $T(au, v) = A(R(au), v)$

$T(au, v) = A(R(au), v)$

$T(u, av) = A(R(au), v)$ (T2a)

$A(R(u), av) = A(R(au), v)$ [IH] (On the left hand side. To learn why it's ok to use our induction hypothesis even though we have $T(u, av)$ instead of $T(u, v)$, see the 'note' about our IH above).

$A(R(u), av) = A(A(R(u), a\bullet), v)$ (R2a, since it's the only way we can simplify)

$A(R(u), av) = A(R(u), A(a\bullet, v))$ (Lemma 1, since I'd like to get av on both sides)

$A(R(u), av) = A(R(u), aA(\bullet, v))$ (A2a)

$A(R(u), av) = A(R(u), av)$ (A1)

Induction Step 2b: Show $T(bu, v) = A(R(bu), v)$

This is similar to Step 2a above.

3) Answer the following questions about trees.

3a) Draw all the non-isomorphic trees (not necessarily binary trees) of size 3.

There are only two of these. If you draw the graph of the relation T1 = $\{(a, b)(b, c)\}$ that's one of them. The other is the graph of the relation T2 = $\{(a, b)(a, c)\}$. Any other trees with three nodes will be isomorphic to one of these two.

3b) Draw all the non-isomorphic *binary* trees of depth 2.

The depth of a tree is the length of the longest path, in edges, from the root of the tree to a leaf. There are two different binary trees of depth 2. One is the graph of the relation T1 = $\{(a, b), (a, c), (b, d), (b, e)\}$, and the other is the graph of the relation T2 = $\{(a, b), (a, c), (b, d), (b, e), (c, f), (c, g)\}$.

4) The language T which is defined below represents the set of binary trees. Use it to answer the following questions.

	$T \subset \{n, l,), (\}^+$
1.	$l \in T$
2.	$t_1, t_2 \in T \Rightarrow (n\ t_1\ t_2) \in T$
3.	There is nothing else in T.

4a) Which of these strings are in the language T?

Ok (i)	l	No (iv)	(l)
Ok (ii)	(n l l)	No (v)	(n n l)
Ok (iii)	(n (n l l) l)	Ok (vi)	(n (n l l) (n l (n l l)))

4b) Fill in the blanks to complete the following recursive functions:

	$LeafCount : T \to N$ (LeafCount(t) = the number of l's (leaves) in t)
L1.	$LeafCount(l) = 1$ (There's 1 l in l)
L2.	$LeafCount((n\ t_1\ t_2)) = LeafCount(t_1)\ +\ LeafCount(t_2)$ (All the l's in t_1 and in t_2)

	$IntNodes : T \to N$ (IntNodes(t) = the number of n's (interior nodes) in t)
N1.	$IntNodes(l) =\ 0$ (There are no n's in l)
N2.	$IntNodes((n\ t_1\ t_2)) = IntNodes(t_1)\ +\ IntNodes(t_2)\ \ + 1$ (All the n's in t_1 and t_2, plus the one we added at the front)

4c) Prove that for all $t \in B$, $LeafCount(t) = IntNodes(t) + 1$

Base Case: $t = l$

$LeafCount(l) = IntNodes(l) + 1$

$1 = IntNodes(l) + 1$ (L1)

$1 = 0 + 1$ (N1)

$1 = 1$

Induction Hypothesis

(since it takes both t_1 and t_2 to make $(n\ t_1\ t_2)$, we need to have two induction hypotheses)

Assume: $LeafCount(t_1) = IntNodes(t_1)+1$ and $LeafCount(t_2) = IntNodes(t_2)+1$

Induction Step

Show: $LeafCount((n\ t_1\ t_2)) = IntNodes((n\ t_1\ t_2)) + 1$

$LeafCount(t_1)\ + LeafCount(t_2) = IntNodes((n\ t_1\ t_2)) + 1$ (L2)

$(IntNodes(t_1) + 1) + (IntNodes(t_2) + 1) = IntNodes((n\ t_1\ t_2)) + 1$ [IH] (both of them)

$(IntNodes(t_1) + 1) + (IntNodes(t_2) + 1) = (IntNodes(t_1) + IntNodes(t_2) + 1) + 1$ (N2)

$IntNodes(t_1) + IntNodes(t_2) + 2 = IntNodes(t_1) + IntNodes(t_2) + 2$

5) Given $P \equiv x+(y-z)+wx$, write the results of the following substitutions:

a) $P \begin{bmatrix} a, & b \\ x, & y \end{bmatrix} = a + (b - z) + wa$

b) $P \begin{bmatrix} b \\ q \end{bmatrix} = x + (y - z) + wx$

c) $P \begin{bmatrix} y, & b \\ x, & y \end{bmatrix} = y + (b - z) + wy$

d) $P \begin{bmatrix} y \\ x \end{bmatrix} \begin{bmatrix} b \\ y \end{bmatrix} = (y + (y - z) + wy) \begin{bmatrix} b \\ y \end{bmatrix} = (b + (b - z) + wb)$

e) $P \begin{bmatrix} (y-z) \\ x \end{bmatrix} \begin{bmatrix} (y-z) \\ w \end{bmatrix} = ((y - z) + (y - z) + w(y - z)) \begin{bmatrix} (y-z) \\ w \end{bmatrix} = ((y - z) + (y - z) + (y - z)(y - z))$

6) Answer the following questions about the program verification problem below (x != 0 is another way of writing $x \neq 0$):

```
{x = A}
z := B;
while (x != 0) {x + z = A + B}
   if  (x > 5)
       then begin  z := z + 5;  x := x - 5;  end
       else begin  x := x - 1;  z := z + 1; end
end
{z = A+B}
```

6a) Is this an appropriate outline of the verification proof for this program? Correct any mistakes in the outline below:

Label the invariant: $I \equiv \{x + z = A + B\}$

This is fine. This is just so we can use I to mean $\{x + z = A + B\}$ below.

Step 1: Prove I $\left[\begin{array}{cc} A, & B \\ x, & z \end{array}\right]$ **is true.**

This is fine. The first step is to prove that I is true at the beginning of the loop, with the initial values. In this program the initial values are $x = A, z = B$, so step 1 is proving I is true with the substitutions A for x and B for z.

Step 2a: Prove $I \wedge (x > 5) \Rightarrow I \left[\begin{array}{cc} A-5, & B+5 \\ x, & z \end{array}\right]$

This is Wrong! The second step is to prove that if I is true before executing the statements in the loop body, I will still be true after those statements have been executed. Since we have an 'if' in the loop body we need two cases, and our Step 2a is meant to take care of the $(x > 5)$ case. So if I is true before we execute: "then begin z := z + 5; x := x - 5; end", we want it to still be true afterwards using the new values for z and x. Which are *not* $A - 5$ and $B + 5$ except in the very special case that you happen to be looking at the first iteration of the loop. We need our invariant to hold for any iteration though! So what we *should* prove here is: $I \wedge (x > 5) \Rightarrow I \left[\begin{array}{cc} x-5, & z+5 \\ x, & z \end{array}\right]$.

Step 2b: Prove $I \wedge (x \leq 5) \Rightarrow I \left[\begin{array}{cc} x-1, & z+1 \\ x, & z \end{array}\right]$

Fine. This handles the 'else' case of step 2 correctly.

Step 3: Prove $I \wedge (x \leq 5) \Rightarrow \{z = A + B\}$

Close. At the end of the program we can use the fact that I must still be true to help prove that our conclusion is true–that the program ends with $\{z = A + B\}$. But we also have another very important piece of information to help us prove the conclusion, and it's not $(x \leq 5)$. In order to reach the end of the program, we must exit the loop, which means our loop condition $(x \neq 0)$ must be false. So in addition to using the fact that I is true at the end of the program, we can use the fact that $x = 0$ at the end of the program to prove that $\{z = A + B\}$. Step 3 should be: $I \wedge (x = 0) \Rightarrow \{z = A + B\}$

6b) Complete the verification proof for the program above.

Label the invariant: $I \equiv \{x + z = A + B\}$

Step 1: Prove I $\left[\begin{array}{cc} A, & B \\ x, & z \end{array}\right]$ **is true.**

I $\left[\begin{array}{cc} A, & B \\ x, & z \end{array}\right] = \{A + B = A + B\}$ Which is true.

Step 2a: Prove $I \wedge (x > 5) \Rightarrow I \left[\begin{array}{cc} x-5, & z+5 \\ x, & z \end{array}\right]$

So we assume I is true–we assume that $(x + z = A + B)$... and we also know $(x > 5)$ (although we don't end up using that fact on this proof)

We need to show $I \left[\begin{array}{cc} x-5, & z+5 \\ x, & z \end{array}\right]$ is true.

$I \left[\begin{array}{cc} x-5, & z+5 \\ x, & z \end{array}\right] = \{(x-5) + (z+5) = A + B\}$

$\{(x-5) + (z+5) = A + B\}$ can be simplified to $\{x + z = A + B\}$, which is what we assumed was true, so we're done.

Step 2a: Prove $I \wedge (x \geq 5) \Rightarrow I \left[\begin{array}{cc} x-1, & z+1 \\ x, & z \end{array}\right]$

So we assume I is true: $(x + z = A + B)$... and we also know $(x \geq 5)$ (although we don't end up using that fact on this proof)

We need to show $I \left[\begin{array}{cc} x-1, & z+1 \\ x, & z \end{array}\right]$ is true.

$I \left[\begin{array}{cc} x-1, & z+1 \\ x, & z \end{array}\right] = \{(x-1) + (z+1) = A + B\}$

$\{(x-1) + (z+1) = A + B\}$ can be simplified to $\{x + z = A + B\}$, which is what we assumed was true, so we're done.

Step 3: Prove $I \wedge (x = 0) \Rightarrow \{z = A + B\}$

If I is true, we know that $x+z = A+B$, and if $x = 0$, then that means $0+z = A+B$, which simplifies to $z = A + B$.

If you'd like to know more about loop invariants, see the review problem from the second midterm.

7) Prove that for all $n \geq 0$, $t_n = a + \frac{n^2+n}{2}$, given the following recursive definition of t_i:

$$t_0 = a$$

$$\vdots$$

$$t_{k+1} = t_k + k + 1$$

This was actually on the last midterm review sheet. There's a link to the solution from the website.

If you have any questions at all over this material, you can feel free to email the AI at ctask@indiana.edu... (although there's no guarantees for emails received after noon on friday).