CS311: Homework #2

Due on September 23, 2013 at 4:30pm

Professor Lathrop Section 3

Josh Davis

Problem 1

Using a loop invariant, prove that the following property holds for MAX:

$$\forall 0 \leq i < n,\ list[i] \leq MAX(list)$$

Solution

Proof. To prove the following property holds, we will use the following loop invariant:

> When looping, of all of the elements in the range $list[0 \ldots i]$, $maxValue$ will contain the the value that is the largest element in the range of elements.

By proving the loop invariant, it will prove the correctness of the function $MAX(list)$. Thus the property will hold.

Initialization
At the beginning of the loop, $i = 0$. Since $maxValue = list[0]$, there is only one value in the range of elements. It is trivially true that the max value of a range with 1 element is always that element. Thus the the loop invariant holds.

Maintenance
During each iteration of the while loop on line 3, the value of $list[i]$ is compared to $maxValue$. If $list[i] > maxValue$, then $maxValue = list[i]$.

Since this happens each loop of the iteration, everytime i is incremented on line 7, the range of elements consists of $list[0 \ldots i]$. Thus since i is being incremented by one, the range is growing by one everytime. Since $maxValue$ is the largest element in the range $list[0 \ldots i-1]$, we just need to compare the value at $list[i]$ to see if we need to update $maxValue$.

This is exactly what the loop invariant says, thus it proves the maintenance step of the loop invariant.

Termination
When the loop terminates, it terminates when $i < n$. Since i is incremented every iteration in the while loop on line 7, when $i = n$ the loop will stop.

Since n is the length of the array, it is easy to tell that when $i = n$, all items in $list$ have been iterated over and thus $maxValue$ will be set to the maximum value in the whole array from lines 4–5.

Since this corresponds to the termination step of the loop invariant, it is easy to see that the loop invariant holds. □

Problem 2

Prove that a tree with n vertices has precisely $n-1$ edges.

Proof. To prove that a tree with n vertices has precisely $n-1$ edges, we will use induction.

Base
First consider the simplest possible tree, a tree with $n = 0$ vertices. Since the tree consists of just one vertex, there are no edges to other vertices and a tree cannot have a cycle. The number of edges is 0 or $n - 1 = 0$. Thus it holds for the base case.

Step
We want to prove that for a tree with $n+1$ vertices, the tree will also have $(n+1)-1$ edges.

Consider a tree, T_0 with n vertices. We can create a new tree, T by taking T_0 and adding one vertex to it. This would give $n+1$ vertices. Because we'd need to connect the vertice to T_0, the number of edges would then be:

$$\text{edges} = T + 1$$

We know this because a tree must be connected and have no cycles. By assuming the induction hypothesis:

$$\begin{aligned}\text{edges} &= T + 1 \\ &= (n-1) + 1 \\ &= (n+1) - 1\end{aligned}$$

Thus we have shown that a tree with $n+1$ vertices has exactly $(n+1)-1$ edges. □

Problem 3

Prove that $6n^4 + 2n^3 + n + 12 \in \Theta(n^4)$.

Proof. To prove that $6n^4 + 2n^3 + n + 12 \in \Theta(n^4)$, we must show the following:

$$\exists c_1 \exists c_2 \forall n \geq n_0,\ c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$$

For the first inequality, it is easy to see that $n^4 \leq c_1 \cdot (6n^4 + 2n^3 + n + 12)$ even for $c_1 = 1$ and $n_0 = 1$.

Now we must show the second part of the inequality:

$$\begin{aligned}6n^4 + 2n^3 + n + 12 = & \\ &\leq 6n^4 + 2n^4 + n^4 + 12n^4 \\ &= 24n^4 \\ &\leq c_2 \cdot n^4\end{aligned}$$

when $c_2 = 24$ and $n_0 = 1$.

Therefore, since we have found our two constants, $c_1 = 1$ and $c_2 = 24$, and constrained $n_0 = 1$, we have proven the proposition. Thus the proof is complete.

□

Problem 4

Prove a polynomial of degree k, $a_k n^k + a_{k-1} n^{k-1} + \ldots + a_1 n^1 + a_0 n^0$ is a member of $\Theta(n^k)$ where $a_k \ldots a_0$ are nonnegative constants.

Proof. To prove that $a_k n^k + a_{k-1} n^{k-1} + \ldots + a_1 n^1 + a_0 n^0$, we must show the following:

$$\exists c_1 \exists c_2 \forall n \geq n_0,\ c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$$

For the first inequality, it is easy to see that it holds because no matter what the constants are, $n^k \leq a_k n^k + a_{k-1} n^{k-1} + \ldots + a_1 n^1 + a_0 n^0$ even if $c_1 = 1$ and $n_0 = 1$. This is because $n^k \leq c_1 \cdot a_k n^k$ for any nonnegative constant, c_1 and a_k.

Taking the second inequality, we prove it in the following way. By summation, $\sum_{i=0}^{k} a_i$ will give us a new constant, A. By taking this value of A, we can then do the following:

$$\begin{aligned} a_k n^k + a_{k-1} n^{k-1} + \ldots + a_1 n^1 + a_0 n^0 = & \\ & \leq (a_k + a_{k-1} \ldots a_1 + a_0) \cdot n^k \\ & = A \cdot n^k \\ & \leq c_2 \cdot n^k \end{aligned}$$

where $n_0 = 1$ and $c_2 = A$. c_2 is just a constant. Thus the proof is complete.

□

Problem 5

Give an appropriate positive constant c such that $f(n) \leq c \cdot g(n)$ for all $n > 1$.

1. $f(n) = n^2 + n + 1$, $g(n) = 2n^3$
2. $f(n) = n\sqrt{n} + n^2$, $g(n) = n^2$
3. $f(n) = n^2 - n + 1$, $g(n) = n^2/2$

Solution
We solve each solution algebraically to determine a possible constant c.

Part One

$$\begin{aligned} n^2 + n + 1 = & \\ & \leq n^2 + n^2 + n^2 \\ & = 3n^2 \\ & \leq c \cdot 2n^3 \end{aligned}$$

Thus a valid c could be when $c = 2$.

Part Two

$$\begin{aligned} n^2 + n\sqrt{n} = & \\ & = n^2 + n^{3/2} \\ & \leq n^2 + n^{4/2} \\ & = n^2 + n^2 \\ & = 2n^2 \\ & \leq c \cdot n^2 \end{aligned}$$

Thus a valid c is $c = 2$.

Part Three

$$\begin{aligned} n^2 - n + 1 = & \\ & \leq n^2 \\ & \leq c \cdot n^2/2 \end{aligned}$$

Thus a valid c is $c = 2$.

Problem 6

Find $f(n)$ and $g(n)$ that satisfy the following relationships. Write "None" if neither exist.

1. $f(n) = o(g(n))$ and $f(n) \neq \Theta(g(n))$
2. $f(n) = \Theta(g(n))$ and $f(n) = o(g(n))$
3. $f(n) = \Theta(g(n))$ and $f(n) \neq O(g(n))$
4. $f(n) = \Omega(g(n))$ and $f(n) \neq O(g(n))$

Solutions

Part One $f(n) = n$ and $g(n) = n^2$. $f(n)$ is dominated by $g(n)$ yet $g(n)$ isn't a tight bound on $f(n)$.

Part Two None. Any function $f(n)$ that is tightly bounded by $g(n)$ cannot be dominated by the same function.

Part Three None. If $f(n)$ is bounded tightly with $g(n)$ (Big Theta definition), then there is no way it isn't bounded asymptotically by $g(n)$ which is what Big Oh means. Big Theta is essentially a stronger statement than Big Oh and thus a subset.

Part Four $f(n) = n^2$ and $g(n) = n$. Big Theta is a lower bound thus $f(n)$ is asymptotically greater than $g(n)$. However it doesn't say anything about $f(n)$'s upper bound.

Problem 7

Prove that if $f_1(n) \in O(g_1(n))$ and $f_2(n) \in O(g_2(n))$, then $f_1(n) + f_2(n) \in O(g_1(n) + g_2(n))$.

Proof. To prove the above proposition, we must show that there exists a c and n_0 such that:

$$f_1(n) + f_2(n) \leq c \cdot (g_1(n) + g_2(n))$$

Since $f_1(n) \in O(g_1(n))$, there exists a c_1 and n_1. Also, since $f_2(n) \in O(g_2(n))$, there exists a c_2 and n_2.

$$\begin{aligned} f_1(n) + f_2(n) &\leq c_1 \cdot g_1(n) + c_2 \cdot g_2(n) \\ &\leq (c_1 + c_2) \cdot g_1(n) + (c_1 + c_2) \cdot g_2(n) \\ &= (c_1 + c_2) \cdot (g_1(n) + g_2(n)) \\ &\leq c \cdot (g_1(n) + g_2(n)) \end{aligned}$$

Since $c = c_1 + c_2$, it is just a constant. By choosing n_0 in a similar fashion, it is easy to see that the propsition holds and thus the proof is complete. □

Problem 8

Suppose $f_1 \in \Theta(g_1)$ and $f_2 \in \Theta(g_2)$. Prove that $f_1/f_2 \in \Theta(g_1/g_2)$.

Proof. To prove the above proposition, we must show that there exists a c_0, c_1 and n_0 such that:

$$c_0 \cdot \frac{g_1(n)}{g_2(n)} \leq \frac{f_1(n)}{f_2(n)} \leq c_1 \cdot \frac{g_1(n)}{g_2(n)}$$

Since $f_1(n) \in \Theta(g_1(n))$, there exists a c_2, c_3 and n_1. Also, since $f_2(n) \in \Theta(g_2(n))$, there exists a c_4, c_5 and n_2.

For the first part of the inequality:

$$\begin{aligned}\frac{c_2 \cdot g_1(n)}{c_4 \cdot g_2(n)} &= \left(\frac{c_2}{c_4}\right) \cdot \frac{g_1(n)}{g_2(n)} \\ &= c_0 \cdot \frac{g_1(n)}{g_2(n)} \\ &\leq \frac{f_1(n)}{f_2(n)}\end{aligned}$$

For the second part of the inequality:

$$\begin{aligned}\frac{f_1(n)}{f_2(n)} &\leq \frac{c_3 \cdot g_1(n)}{c_5 \cdot g_2(n)} \\ &= \left(\frac{c_3}{c_5}\right) \cdot \frac{g_1(n)}{g_2(n)} \\ &\leq c_1 \cdot \frac{g_1(n)}{g_2(n)}\end{aligned}$$

Since c_0, c_1 are both constants, we have shown that the proposition holds. Thus the proof is complete. □

Problem 9

Suppose $f(n) \in O(g(n))$.

Part One
Prove that $f(n) + g(n) \in O(g(n))$.

Proof. We will show that $f(n) + g(n) \in O(g(n))$ by using the definition of Big-Oh. Suppose that $f(n) \in O(g(n))$. That means $\exists c_0 \exists n_0 \forall n \geq n_0,\ f(n) \leq c_0 \cdot g(n)$.

Now we must just show that $\exists c_1 \exists n_0 \forall n \geq n_0\ f(n) + g(n) \leq c_1 \cdot g(n)$.

By taking $f(n) \leq c \cdot g(n)$, we can add $g(n)$ to both sides giving us:

$$f(n) + g(n) \leq c \cdot g(n) + g(n)$$

Since c is just a constant, this gives us the following:

$$f(n) + g(n) \leq (c + 1) \cdot g(n)$$

Which means our n_0 stays the same but our constant just becomes $c_1 = c_0 + 1$. Since this proves what we wished to prove, our proof is complete. □

Part Two
Prove that $O(f(n) + g(n)) = O(g(n))$.

Proof. To prove that $O(f(n) + g(n)) = O(g(n))$, we need to show that $O(f(n) + g(n)) \subseteq O(g(n))$ and $O(g(n)) \subseteq O(f(n) + g(n))$ because Big-Oh notation creates a set of functions.

First, we must show that $O(f(n) + g(n)) \subseteq O(g(n))$ We can do this by proving that for every $h(n) \in O(f(n) + g(n))$, it will exist in $O(g(n))$ or more formally: there exists a c and n_0 such that for all $n \geq n_0$, $h(n) \leq c \cdot g(n)$.

Suppose that $f(n) \in O(g(n))$ and $h(n) \in O(f(n) + g(n))$. This means that for $f(n)$, there exists a c_0 and n_1 such that for all $n \geq n_1$, $f(n) \leq c_0 \cdot g(n)$. This also means that for $h(n)$, there exists a c_1 and n_2 such that for all $n \geq n_2$, $h(n) \leq c_1 \cdot (f(n) + g(n))$.

Using this, we get the following:

$$\begin{aligned} h(n) &\leq c_1 \cdot ((f(n) + g(n))) \\ &\leq c_1 \cdot (c_0 \cdot g(n) + g(n)) \\ &= c_1 c_0 \cdot g(n) + c_1 \cdot g(n) \\ &= (c_1 c_0 + c_1) \cdot g(n) \\ &\leq c \cdot g(n) \end{aligned}$$

If we just let $c = c_1 c_0 + c_1$ and $n_0 = n_2 n_1 + n_2$, then we can see that $h(n) \in O(g(n))$ and the first part of the proof is complete.

Problem 9 continued on next page...

Second, we must show that $O(g(n)) \subseteq O(f(n) + g(n))$ We can do this by proving that for every $h(n) \in O(g(n))$, it will exist in $O(f(n) + g(n))$, or more formally: there exists a c and n_0 such that for all $n \geq n_0$, $h(n) \leq c \cdot (f(n) + g(n))$

Suppose that $f(n) \in O(g(n))$ and $h(n) \in O(g(n))$. This means that for $f(n)$, there exists a c_0 and n_1 such that for all $n \geq n_1$, $f(n) \leq c_0 \cdot g(n)$. This also means that for $h(n)$, there exists a c_1 and n_1 such that for all $n \geq n_2$, $h(n) \leq c_1 \cdot g(n)$.

Using this, we get the following:

$$\begin{aligned} h(n) &\leq c_1 \cdot g(n) \\ &\leq (c_0 + c_1) \cdot g(n) \\ &= c_0 \cdot g(n) + c_1 \cdot g(n) \\ &\leq c_0 \cdot f(n) + c_1 \cdot g(n) \\ &\leq (c_0 + c_1) \cdot (f(n) + g(n)) \\ &\leq c \cdot (f(n) + g(n)) \end{aligned}$$

If we let $c = c_0 + c_1$ and $n_0 = n_1 + n_2$, then we can see that $h(n) \in O(f(n) + g(n))$. Thus since we have proven each side of the equality of sets, our proof is complete. $\square$

Problem 10

Prove or disprove: Big Oh defines an equivalence relation on the set of all functions: $f : \mathbb{N} \to \mathbb{R}$.

Counterexample
Let $f(n) = n$ and $g(n) = n^2$.

According to the definition of an equivalence relation, the relation creates essentially partitions where each element belongs to one and only one partition.

More formally, the properties that this implies are the following:

1. Reflexivity: aRa.
2. Symmetry: if aRb, then bRa.
3. Transitivity: if aRb and bRc, then aRc.

Using the second property of an equivalence relation, it is easy to see that Big-Oh isn't an equivalence relation because since $n \in O(n^2)$, it should follow that $n^2 \in O(n)$, but we know that isn't the case. Thus it is a counterexample and the proposition is false.

Problem 11

Cover a grid of squares with 2^n rows and 2^n columns, which means $(2^n)^2$ squares in the grid. The only thing you can use to cover it is L-shaped pieces.

Prove it is possible to cover for any n using induction.

Proof. **Base**
To prove the base case, we will prove that a grid with squares can be covered when $n = 0$ and $n = 1$ for comprehensiveness.

When $n = 0$, the case is trivial. The number of rows equals $2^n = 2^0 = 1$, the columns also equal $2^n = 2^0 = 1$. This is just a grid with one tile. This requires 0 L-shaped pieces. Only one square is uncovered and there are no overlaps. The adversary must then choose that square. Thus the base when $n = 0$ is satisfied.

When $n = 1$, the number of rows and columns both are $2^n = 2^1 = 2$. This is just a square with 4 total squares.

Our adversary can then has four options of which to choose. It can either choose the top left, top right, bottom left, or bottom right.

Without loss of generality, in all cases, the square that is chosen can be left uncovered by placing just one L-shaped piece such that it is rotated leaving three squares covered and the chosen square uncovered. Thus it can be seen that when $n = 1$, the base case is still satisfied.

Step
In the step, we can assume that for n the problem of covering the grid of $(2^n)^2$ is solvable. By assuming that, we will show that the problem is also solvable when $n = n + 1$, thus completing the induction and proving the problem.

When $n = n+1$, that means there will be $(2^{n+1})^2$ squares in the grid. This means when the adversary picks his square, there will be $(2^{n+1})^2 - 1$ squares. Since there are 3 squares in the L-shaped piece:

$$\begin{aligned}(2^{n+1})^2 - 1 &= (2^{n+1}) \cdot (2^{n+1}) - 1 \\ &= 4^{n+1} - 1 \\ &= 4 \cdot 4^1 \cdot 4^2 \ldots 4^{n-1} \cdot 4^{n+1} - 1 \\ &= (4-1) \cdot 4^1 \cdot 4^2 \ldots 4^n \cdot 4^{n+1} \\ &= 3 \cdot 4^1 \cdot 4^2 \ldots 4^n \cdot 4^{n+1} \\ &= 3k\end{aligned}$$

where $k \in \mathbb{Z}$. Thus it is divisible by three meaning that we can completely cover the grid of squares with our L-shape tiles.

The grid also contains 4^n 4-by-4 grids. By greedily covering it starting with the square the adversary picks, it will completely cover the grid. Thus the proposition holds for $n = n + 1$ and the proof by induction is complete. □