Discussion Solutions Week 5

CS 173: Discrete Structures

Tuesday

Problem 1. from Trees and Grammars

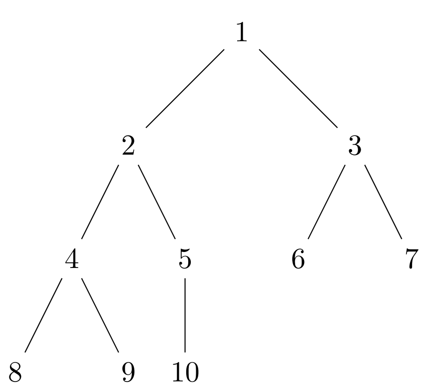
(a) See the diagram below for the tallest possible tree with 10 nodes.

1
|
2
|
3
|
4
|
5
|
6
|
7
|
8
|
9
|
10

  (i) This tree has height 9.

  (ii) The number of internal nodes is 9, since internal nodes include all nodes with at least one child node. That leaves one leaf node.

(b) See the diagram below for the shortest possible tree with 10 nodes.



  (i) The height of this tree is 3.

(ii) This tree has 5 internal nodes and 5 leaves.

(c) No you cannot. A full binary tree requires that each node has either 0 or 2 children. In the diagram above, nodes 1 through 4 have two children each, nodes 6 through 10 have zero children, but node 5 only has one child. There is nowhere to move node 10 to avoid having one node with a single child.

(d) The number of nodes n in a full binary tree must be of the form$n = 2m + 1$ for some $m \in \mathbb{N}$.

Problem 2. from Trees and Grammars

(a) The following grammar will generate all palindromes consisting of "a"s and "b"s with start symbol S and terminal symbols a and b:

$$S \to aSa \mid bSb \mid b \mid a \mid \epsilon$$

(Commentary: if you forget the $S \to a \mid b$ rules, the grammar will only generate even-length palindromes, and if you forget $S \to \epsilon$ rule, the grammar will only generate odd-length palindromes.)

(b) Here is one way to write the grammar:

$$\begin{aligned} S &\to aSa \mid B \mid a \\ B &\to bBb \mid b \mid \epsilon \end{aligned}$$

Wednesday

Problem 13.3. in Discussion Manual

(b) Let T be a parity tree; we will prove T has the parity property by induction on its height h.

Base: For height 0, T is just a solitary root. That root is also a leaf so it is orange by rule 1 of parity trees. Thus there is an odd number of leaves (1) and the root is orange, so T has the parity property.

(Commentary: You might think you need two base cases here: height 0 for an orange-root case and height 1 for blue-root. However, while including an extra base case doesn't invalidate the proof, it's not actually necessary here - to see that, try following through the logic of the induction step below using the concrete height 1 tree plugged in for T everywhere.)

Induction: Suppose that all parity trees with height less than h have the parity property. Then for parity tree T with height h, consider its left and right subtrees T_ℓ and T_r, and let n_l and n_r be the number of leaves in the respective subtrees. Notice that T_ℓ and T_r are also parity trees, so since they have height smaller than h, by the IH we know they both have the parity property. *(You can* not *say that they have height $h-1$ - one of them definitely does, but the other could be arbitrarily shorter. This is why it is important that we are using a* strong *IH.)* Now we get four cases:

Case 1: n_ℓ and n_r are both even. Then by the parity property, T_ℓ and T_r both have blue roots. Then by rule 2 of parity trees, T also has a blue root. And we know the total number of leaves is $n_\ell + n_r$ which is even (because its the sum of two evens), so T has the parity property.

Case 2: n_ℓ and n_r are both odd. Then by the parity property, T_ℓ and T_r both have orange roots. Then by rule 2 of parity trees, T has a blue root. And we know the total number of leaves is $n_\ell + n_r$ which is even (because its the sum of two odds), so T has the parity property.

Case 3: n_ℓ is even and n_r is odd. Then by the parity property, T_ℓ has a blue root and T_r has an orange root. Then by rule 2 of parity trees, T has an orange root. And we know the total number of leaves is $n_\ell + n_r$ which is odd (because its the sum of an even and an odd), so T has the parity property.

Case 4: n_ℓ is odd and n_r is even. See case 3 with the roles of T_ℓ and T_r reversed.

Thus T has the parity property in every case.

Problem 13.2. in Discussion Manual

(a) Proof by induction on the tree height.

Base: Notice that trees from this grammar always have height at least 1. The only ways to produce a tree of height 1 are the third and fourth rules; in each case the tree ends up with one node labeled a and at most one labeled b.

Induction: Assume that any tree of height less than some $k > 1$ has at least as many a nodes as bs. Now consider a generated tree with height k. The root must be labelled S and the grammar rules that can produce trees of height greater than 1 give us two cases for what the children are:

Case 1: The root's children are labeled a, S, b, and S. Let T_1 and T_2 be the subtrees rooted at the nodes labeled S, and let a_1, a_2, b_1, b_2 be how many a nodes and b nodes are in each subtree. Since T_1 and T_2 have height less than k, the IH applies to them, so $a_1 \geq b_1$ and $a_2 \geq b_2$. Putting these two inequalities together and adding one, we establish that $a_1 + a_2 + 1 \geq b_1 + b_2 + 1$. And $a_1 + a_2 + 1$ is just the total number of a nodes in the tree while $b_1 + b_2 + 1$ is the total number of b nodes, so we have shown that there are at least as many as overall as bs.

Case 2: The root's children are labeled S, a, S. The logic here is exactly like case 1 except with one fewer b node, so there are definitely at least as many as as bs.

Thus in every case there are at least as many as as bs.

Thursday

Problem 1. from Big-O Tutorial Problems

(a) We want to show there are positive reals k, c such that $\forall n \geq k, 0 \leq 2^n \leq c \cdot n!$. Let $k = 4$ and $c = 1$. Then it remains to show that $\forall n \geq 4, 0 \leq 2^n \leq n!$. This follows from Claim 50 in the textbook.

(Commentary: Note that $k = 4$ is not the tightest bound on n. You can attempt to compute the tightest bound by "solving" the inequality $2^n \leq n!$. If it's not clear to you how to do this; try taking the $\log_2$ of both sides and applying log rules. You should end up with a claim that matches $n \geq k$.)

(b) This statement is false. As a counterexample, consider $f(n) = 2^n$ and $g(n) = 1$. Then $f(n)$ is $O(2^n)$ and $g(n)$ is $O(n!)$, but $f(n)$ is not $O(g(n))$. *(Commentary: Informally, "$g(n)$ is $O(n!)$" provides an upper bound on how fast g can grow, but it does not provide a lower bound.)*

Problem 2. from Big-O Tutorial Problems

Fix f, g, h, and assume that $f(n)$ is $O(g(n))$ and $g(n)$ is $O(h(n))$. Then by definition of big-O, there are (positive real) k_0, c_0 such that $\forall n \geq k_0, 0 \leq f(n) \leq c_0 g(n)$, and also k_1, c_1 such that $\forall n \geq k_1, 0 \leq g(n) \leq c_1 h(n)$. Now we want to show there are k, c such that $\forall n \geq k, 0 \leq f(n) \leq ch(n)$.

Let $k = max(k_0, k_1)$ and $c = c_0 c_1$. Then we need to show $\forall n \geq max(k_0, k_1), 0 \leq f(n) \leq (c_0 c_1) \cdot h(n)$. To do this, fix $n \geq max(k_0, k_1)$. Then we have $0 \leq f(n)$ (since $n \geq max(k_0, k_1) \geq k_0$), and also:

$$\begin{aligned}
f(n) &\leq c_0 g(n) && \text{(since } n \geq max(k_0, k_1) \geq k_0) \\
&\leq c_0(c_1 h(n)) && \text{(since } n \geq max(k_0, k_1) \geq k_1) \\
&= (c_0 c_1) \cdot h(n) && \text{(rearrange)}
\end{aligned}$$

Thus, $f(n)$ is also $O(h(n))$.

Problem 14.2. in Discussion Manual

(b) For this problem let's fix $c = 1$ and find the tightest bound on n (*i.e.,* k). When $c = 1$, we have $\frac{x^3+2x}{2x+1} \leq x^2$. This simplifies to $x^3 + 2x \leq 2x^3 + x^2$. We can divide both sides by x and move the terms to the same side and get $x^2 + x - 2 \geq 0$. Factoring, we have $(x+2)(x-1) \geq 0$. This gets us $x \geq -2$ and $x \geq 1$ or $x \leq -2$ and $x \leq 1$. The only feasible option here is that $x \geq 1$. Thus, $c = 1$ and $k = 1$.

(Commentary: To show more concretely that these values work; try setting $x \geq k$ (in this case $k = 1$), and working to get $\frac{x^3+2x}{2x+1} \leq x^2$.)

(d) In this case we will choose c and k upfront and show that the big-O inequality must hold. Let's set $c = 1$ and $k = 3$.

Now, we have $x \geq k$, or $x \geq 3$, and we want to show that in this case, $2^x + 17 \leq 3^x$. Let's rephrase the claim to be $2^x \leq 3^x - 17$. We will show this using induction on x.

Base case: $x = 3$, $2^3 \leq 3^3 - 17$. $8 \leq 10$; the base case holds.

IH: Suppose $2^x \leq 3^x - 17$ for $x = 3..k-1$. Then our goal is to show that $2^k \leq 3^k - 17$.

Let's start with 2^k. This can be written as $2 * 2^{k-1}$. We know that $2^{k-1} \leq 3^{k-1} - 17$ by the IH. Then, $2^k \leq 2 * (3^{k-1} + 17) = 2 * 3^{k-1} - 34$. Using algebra, we can show that $2^k \leq 2 * 3^{k-1} - 34 < 3 * 3^{k-1} - 34 = 3^k - 34 < 3^k - 17$. Thus, we have shown $2^k \leq 3^k - 17$.

Friday

See solution here: https://mfleck.cs.illinois.edu/study-problems/inequality-induction/inequality-induction-1-sol.html