Parallel & Concurrent Programming
Theory Assignment 1: Filter Lock Correctness Spring 2022

Vibhanshu Jain
CS19B1027

In the class, we discuss Filter Lock and its correctness proof. We discussed a modification to Lemma 2.4.1 as shown below. Please prove or disprove the modified lemma. Note that you disprove a statement (or lemma) by giving a counter-example.

Lemma 2.4.1: For j between 1 and n − 1, there are at most n − j + 1 threads at level j.

Proof:
By induction at j. In the best case where j = 0, means the level 0, we have all the threads waiting.
Now, let's check for level 1, with our assumption that there are at most *n-j+1*, we get that there can be a maximum of *n-1+1 = n* threads at this level which is true as all the threads are there at level 1.
Now let's assume that there we are on level j, then there are at most, *n-j+1* threads in this level.
Now following the algorithm/code we can say that at least one thread will be stuck at this level.
Thus in level *j+1*, there can be (maximum number of threads in level j - 1) which is, *n-j+1-1 = n-j*.
After the rearrangement we get, *n-j = n-(j+1)-j*, which is exactly the same as we expected.
Thus this shows that our lemma is correct.
Let's check for the level, n-1 which is just above the critical section, we can have a maximum of *n-(n-1) - 1 = 2*, which means at most two threads can be there. And which true as there can be a maximum of one thread in the level n, the critical, so the level n-1 can have a maximum of two threads.