Which problems did you find in the two models as a result of using LTSA?

Were these problems also present in the Java implementation?

If you find problems with the Java solution submitted for part 1a, Solution 2, why do you think you picked these up now and not before submitting the Java code?

If you did not find problems with your part 1a, Solution 2 submission, were you convinced when you submitted part 1a, Solution 2 that no problems existed?

Why did you believe this? Do you still believe there are no problems?

After implementing the solutions using FSP, deadlocks were found in the solution 1 while no deadlock was found in the solution 2. And no progress violation was found in both of the solutions. The deadlock did present in the Java implementation of the solution 1 in part 1a. However, the problem is not easy to be identified in the Java solution, because the deadlock only occurs when some special input was given.

Although no problems were found after I tested the solution 2 of part 1a, I was not convinced that there was no problem while the part 1a was submitted. Because all the testing can do is to make sure the program works as expected given some test cases or under some conditions. The testing never guarantees that the program is correct, especially when we are dealing with a parallel system with a huge number of possible status.

But, one of the biggest advantages of the LTS analyser is that the analyser will analyse all the possible status of the system looking for the deadlocks and other problems. In my opinion, this will prove the correctness of the system to some extent.

After the FSP implementation, no problems were found in the part 1b solution 2 which is derived from part 1a solution 2 using reverse engineering. So it is reasonable to believe that the model built with FSP is free of deadlocks and property violations. But it is critical to mention that, the FSP implementation is a way to modelling the program in a higher level, which means that, compared to the FSP implementation, the Java implementation have some more detailed design. In other words, while we using the reverse engineering to convert the Java program to the FSP model, some of the information in the Java implementation was lost.

In conclusion, I think it is safe to say that, the general processes of the part 1a, solution 2 should be free of deadlocks. But the differences between the java program and the FSP model may cause some problems that we are not able to foresee in the FSP model.